

# La Teoría de Control y la Gestión Autónoma de Servidores Web

Félix Armando Fermín Pérez

fferminp@unmsm.edu.pe

Lima – Perú

**Resumen:** La adecuada gestión de un sistema de información típico consistente de un servidor de aplicaciones, un servidor web, servicios de mensajería, capas middleware y sistemas operativos, se ha convertido en un gran reto para las organizaciones, debido al coste elevado y la falta de fiabilidad de una administración manual, principalmente cuando existen problemas de congestión y degradación de la calidad del servicio de los servidores. La computación autónoma o autónoma propone mejorar la compleja gestión de los sistemas informáticos dinámicos utilizando la teoría de control realimentado, entre otros. Los sistemas de gestión autónoma, o auto-gestión, se basan en lazos de control que monitorean el ambiente, modelan su comportamiento en ese entorno, y toman acción para modificar dicho ambiente o su propio comportamiento, de manera tal que el administrador humano es reemplazado por un sistema de control automático. Este artículo muestra como caso de estudio a la gestión autónoma de los servidores web.

**Abstract:** The management of a typical information system consisting of an application server, web server, messaging, middleware layers and operating systems, has become a major challenge for organizations due to high cost and unreliability of manual administration especially when there are problems of congestion and degradation of quality of service. Autonomous or autonomic computing aims to improve the management of complex dynamic systems using feedback control theory, among others. Autonomous management systems or self-management systems based on control loops try to monitor the environment, model their behavior in that environment, and take action to modify the environment or their own behavior, so that the human administrator is replaced by an automatic control system. This paper shows the autonomic management of web servers as a study case.

**Palabras clave:** Gestión Autónoma, Teoría de Control, Servidor Web, Computación Autónoma.

## 1. Introducción

Los servidores web, de base de datos y de correo electrónico, se han constituido en componentes vitales de toda organización moderna, por lo que es importante invertir no solo en su implementación, sino también en su administración y mantenimiento. En los servidores web, los administradores de estos sistemas informáticos requieren de bastante experiencia, dedicación y labor manual, además de contar con herramientas software para el monitoreo del tráfico de red y de la utilización de recursos computacionales como la capacidad de procesamiento y la cantidad de memoria utilizada, tal como sucede, por ejemplo, en el caso de las empresas de telecomunicaciones, para asegurar el cumplimiento de los niveles de calidad de servicio pactados dentro de los acuerdos de nivel de servicio, y en el caso de las empresas en general, para garantizar la disponibilidad y buen funcionamiento de sus sistemas de gestión.

La gestión autónoma o autonómica de servidores informáticos es una interesante línea de investigación relativamente reciente [1] que promueve el uso de sistemas de control automático en lazo cerrado, para controlar una determinada variable de salida, por lo general el tiempo de respuesta del servidor, mediante la manipulación adecuada de los niveles de CPU o de memoria empleados, o para lograr los tiempos de respuesta pactados mediante el control de las prioridades de programación, la reasignación de memoria o la reasignación del ancho de banda. En este sentido, existe un creciente interés en reducir la intervención humana identificada como parte principal de los problemas [2], yendo hacia la gestión autónoma o autogestión, que es parte de la computación autónoma [3] o también denominada computación autónoma, a semejanza de

cómo el sistema nervioso autónomo humano regula nuestra temperatura, respiración, ritmo cardíaco y otros sin que estemos siempre conscientes de ello. Generalmente, los profesionales de informática utilizan un procedimiento ad hoc para la administración de servidores donde la gestión de recursos está aún fuertemente supeditada al control y administración manual.

Uno de los pilares de la computación autónoma lo constituye la teoría de control, disciplina sistematizada, basada en las matemáticas, ampliamente utilizada en diversos ámbitos de las ingenierías clásicas, como la ingeniería mecánica, eléctrica, aeronáutica y otros en las que es útil para analizar y diseñar lazos de control, generalmente realimentados, con la finalidad de regular y controlar automáticamente las características de cualquier tipo de sistema [4].

En la literatura especializada, se puede hallar abundante información de la aplicación de la teoría de control en las ingenierías clásicas, pero en menor medida, de su aplicación en sistemas informáticos, como los servidores web, de correo electrónico y de base de datos. Desde inicios de los 90, se estudió su uso en el control automático de redes de comunicación [5] en el protocolo TCP [6] en el control de procesamiento de datos mediante las asignaciones de memoria y de procesador [7] en el control dinámico de la calidad de la señal de vídeo enviada por las líneas de transmisión [8]. Desde el 2000, los estudios se han centrado en la mejora de la performance de los servidores de aplicaciones web, de base de datos y de correo electrónico, mediante el diseño de controladores realimentados [9] [10] que reduzcan la intervención humana. Últimamente, se han utilizado modelos SISO, basados en la medición del tiempo de

respuesta del servidor y el conocimiento del valor del parámetro que determina la máxima cantidad de usuarios concurrentes en el servidor [11], así como en la utilización de técnicas de la inteligencia artificial, específicamente, lógica difusa [12] [13], redes neuronales [14] y algoritmos genéticos en el diseño de los controladores.

El presente artículo, en el que se estudia la relación entre la teoría de control y la gestión autónoma de servidores web, está organizado de manera tal que en la sección 2, se proporcionan conceptos fundamentales de la gestión autónoma y la computación autónoma. La sección 3 plantea una metodología de aplicación de la teoría de control a sistemas informáticos; finalmente, en la sección 4, se describen las conclusiones y sugerencias.

## 2. Conceptos fundamentales

La gestión autónoma forma parte de la computación autónoma, la que a su vez tiene como uno de sus pilares a la teoría de control; a continuación se tratan aspectos básicos antes de describir la metodología a utilizar.

### 2.1 Gestión autónoma

La tecnología influye sobre todos los aspectos de la vida cotidiana. No solo los servicios y aplicaciones son complejos, heterogéneos y dinámicos, sino también la infraestructura de información, como Internet, por ejemplo, que incorpora grandes cantidades de recursos informáticos y de comunicación, almacenamiento de datos y redes de sensores provocando que las infraestructuras informáticas y de información se tornen frágiles, inmanejables e inseguras. Así, en el 2001, IBM formuló un paradigma para lidiar con la complejidad y los problemas evolutivos, la computación autónoma [3], inspirado en el sistema nervioso autónomo humano. Computación autónoma no es solo un nuevo campo de investigación, sino que es la conjunción de teorías y mejores prácticas de áreas, como la teoría de control, algoritmos adaptativos, agentes software, robótica, computación tolerante a fallas, sistemas distribuidos, sistemas en tiempo real, aprendizaje de máquinas, interacción humano-computador, inteligencia artificial y más. También depende del desarrollo y éxito de quienes proporcionan la infraestructura para los sistemas autónomos, como los servicios web y servicios grid, así como de plataformas de arquitectura, como SOA, computación pervasiva y ubicua [15].

El diseño e implementación de sistemas autónomos debe cumplir los siguientes principios básicos:

- Flexibilidad: datos en múltiples plataformas.
- Accesibilidad: siempre disponible.
- Transparencia: ejecuta tareas y se ajusta al usuario.

El objetivo general de la computación autónoma es implementar sistemas y aplicaciones informáticas y software que pueda ser manejada por ellos mismos (self-management) de acuerdo con las políticas de negocio determinada por los humanos. Se caracterizan por su auto configuración (self-configuration), auto curación (self

healing), auto optimización (self-optimization) y auto protección (self-protection) [16]. Ver tabla 1.

Auto configuración	configurarse y reconfigurarse dinámicamente según políticas del negocio, bajo cualquier circunstancia previsible o no.
Auto curación	detectar y reparar problemas de hardware y software.
Auto optimización	siempre buscar la mejora de su performance y eficiencia.
Auto protección	descubrir, identificar y protegerse contra ataque y garantizar su funcionamiento e integridad.

Tabla 1. Propiedades de la gestión autónoma.

En el centro de un sistema autónomo, se halla un sistema de control en lazo cerrado o realimentado, conformado por componentes que actúan juntos para mantener los valores reales del sistema, lo más cercano posible a las especificaciones deseadas.

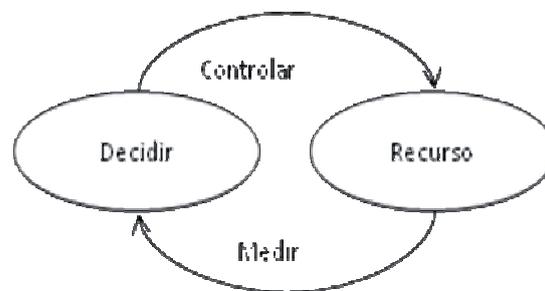


Figura 1. Lazo de control autónomo.

Este lazo de control de la Figura 1 es el elemento de gestión básico que actúa como un administrador de los recursos computacionales mediante el monitoreo, análisis y acciones tomadas sobre un conjunto de políticas de funcionamiento predefinidas del sistema. Estos lazos de control, o elementos autónomos, pueden comunicarse y eventualmente negociar entre ellos y otros tipos de recursos dentro y fuera de la arquitectura autónoma, como se observa en la Figura 2.

El gestor autónomo de la Figura 2 es una implementación que automatiza alguna función administrativa, es un componente que implementa un lazo de control inteligente. Para que un componente del sistema pueda gestionarse a sí mismo, deberá contar con un método automatizado de recolección de detalles de lo que necesita del sistema; analizarlos para determinar si hay algo que modificar; crear un plan o secuencia de acciones que especifique los cambios necesarios y realizar tales acciones. Cuando estas funciones pueden ser automatizadas se forma un lazo de control inteligente o elemento autónomo en la que la teoría de control podría ser de mucha utilidad para el diseño de los gestores autónomos que en su interior poseen un controlador realimentado PID, con lógica difusa, o basado en redes neuronales, por ejemplo. El recurso o elemento administrado es el que va a ser auto gestionado. Este recurso podría ser un simple sistema, o una aplicación

dentro de un sistema, o un grupo de muchos sistemas relacionados lógicamente.

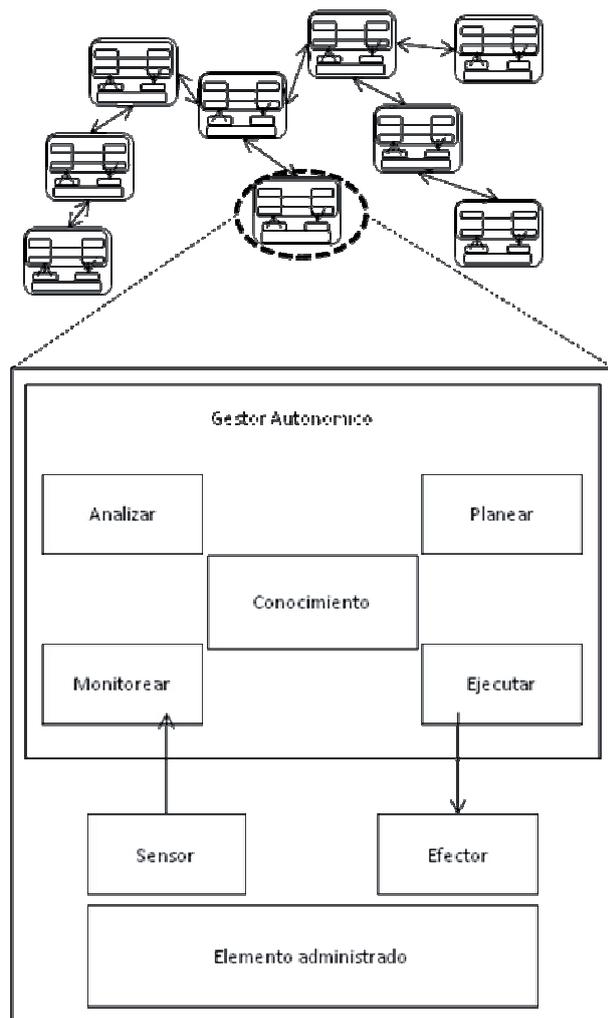


Figura 2. Elemento autónomo. IEEE 2003.

## 2.2 Teoría de Control

La teoría de control realimentado es el mecanismo básico mediante el cual los sistemas de cualquier tipo, sean éstos, mecánicos, eléctricos, o biológicos, mantienen su

equilibrio u homeostasis [17]. Ogata [18] menciona que la teoría de control trata de controlar, regular, gobernar, comandar automáticamente las características estáticas y dinámicas de funcionamiento de sistemas de cualquier tipo. Asimismo, Astrom [19] establece que lo mágico de la realimentación es que se puede crear un sistema que funcione bien con componentes que funcionan pobremente, además de hacerlo insensible a ruidos o perturbaciones, y a variaciones en los componentes que constituyen el sistema en estudio.

La importancia de la aplicación de la teoría de control es que proporciona un enfoque sistemático que se fundamenta en las matemáticas, para diseñar sistemas de control en lazo cerrado, que tengan como principales características el ser estables, evitar fuertes oscilaciones, ser exactos en los valores de salida deseados, y en responder rápidamente a los cambios de la carga de trabajo, estabilizándose en corto tiempo alrededor del valor final del estado estacionario.

Utiliza conceptos tales como el de función de transferencia, que relaciona la salida y entrada de un sistema en particular para estudiar sus propiedades. Estudia la estabilidad de un sistema mediante la determinación de la convergencia de la señal de la salida del sistema cuando la entrada también lo es. Se asocia al concepto de exactitud en el logro del valor planteado de la señal de salida del sistema cuando se aproxima lo más posible al valor de la señal de entrada, aunque en realidad lo que se mide generalmente para fines de control es el denominado error de estado estacionario, definida como la diferencia entre el valor de referencia, lo que se desea, y el valor real de la salida, lo que en realidad se obtiene. Otras características de la aplicación de la teoría de control tiene que ver con, por ejemplo, el tiempo de establecimiento que muestra lo rápido que converge el sistema en estudio, o el tiempo de subida que muestra la velocidad de respuesta, y están relacionadas con el comportamiento temporal del sistema [2].

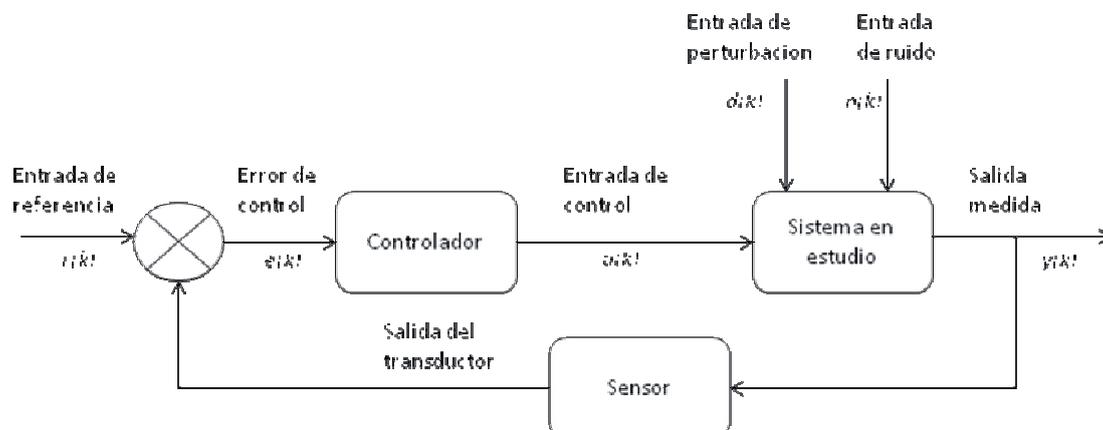


Figura 3. Sistema de control realimentado [4].

Por lo general, como se muestra en la Figura N° 3, al sistema en estudio se le agrega un elemento denominado “controlador”, que dinámicamente ajusta el comportamiento de uno o más elementos basados en los

valores de las mediciones tomadas a la variable de salida del mismo sistema en estudio. En resumen, lo que se hace es manipular el error de control  $e(k)$  mediante algún algoritmo implementado en el controlador, de manera que

éste emita una señal denominada entrada de control  $u(k)$  hacia el sistema en estudio para así lograr la salida deseada  $y(k)$ , que debería ser igual a la entrada de referencia  $r(k)$ , esto es, ser igual al objetivo planteado, cuando la entrada de ruido  $n(k)$  y la entrada de perturbación  $d(k)$  tratan de afectarla negativamente.

En el caso de los sistemas informáticos, la aplicación de la teoría de control a servidores web, está orientado a brindar una alternativa de administración automática de estos servidores, mediante el uso de controladores, que reduzcan y aligeren la intervención humana especializada al tratar de mantenerlos en un rango aceptable de funcionamiento, en eventos de congestión, por ejemplo, que degradan la calidad del servicio o incrementan el coste de operación [1]. Cuando los usuarios manifiestan que observan largos tiempos de respuesta durante sobrecargas que congestionan el acceso al servidor web. Estos lentos tiempos de respuesta no necesariamente dependen de pequeños anchos de banda, o de conexiones demasiado lentas de los usuarios, sino que a menudo el cuello de botella es el mismo servidor web, por lo cual se plantea el uso de controladores realimentados PID, e incluso basados en técnicas de inteligencia artificial, como la lógica difusa y redes neuronales [12] [13] [14], para gestionar automática y autónomamente tales servidores, disminuyendo la intervención humana.

### 3. Metodología

La metodología de aplicación de la teoría de control realimentado a sistemas informáticos, mostrada en la Figura 4, consta de dos fases denominadas en general como fase de identificación de sistemas y fase de diseño del controlador realimentado para el sistema en estudio.

Para hallar el modelo matemático de un servidor informático, se puede emplear dos enfoques: uno basado en leyes y principios básicos y otro fundamentado en un enfoque empírico, denominado identificación de sistemas, que emplea datos recopilados del funcionamiento del mismo sistema.

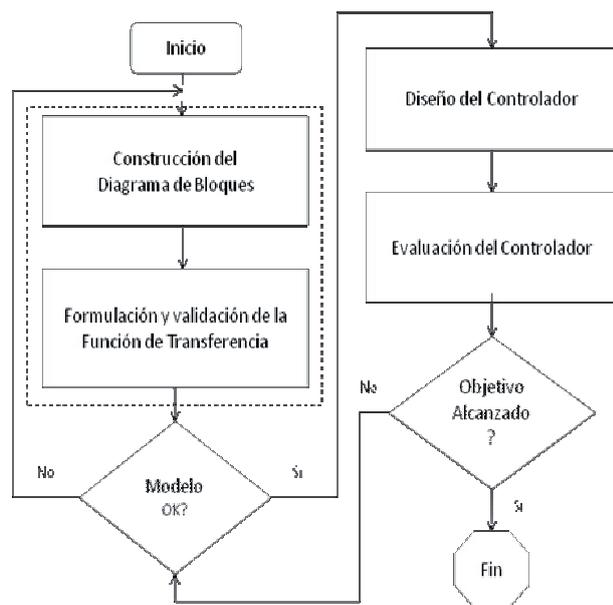


Figura 4. Metodología de aplicación [4].

En trabajos previos se ha tratado de utilizar principios, axiomas, postulados, leyes o teorías básicas para realizar el modelado matemático de sistemas informáticos, pero lamentablemente existen algunos serios inconvenientes con este enfoque. Primero, es difícil construir un modelo bajo estos principios básicos, ya que a menudo se asumen supuestos irreales debido a la naturaleza compleja y estocástica de un sistema informático. Segundo, es necesario tener un conocimiento detallado del sistema en estudio, más aun, cuando cada cierto tiempo se van liberando al mercado actualizaciones del software, por lo que debería contarse con un experto en el comportamiento de servidores de manera permanente. Y tercero, que este enfoque no aborda lo referido a la validación del modelo [10].

La identificación de sistemas es un enfoque empírico donde deben identificarse los parámetros de entrada y salida del sistema en estudio, para luego construir un modelo paramétrico, como el ARX por ejemplo, basándose en técnicas estadísticas de autoregresión. Este modelo o ecuación paramétrica (1) relaciona los parámetros de entrada y de salida del servidor web de acuerdo la siguiente ecuación:

$$y(k+1) = Ay(k) + Bu(k) \quad (1)$$

donde  $y(k)$  : variable de salida  
 $u(k)$  : variable de entrada  
 $A, B$  : parámetros de autoregresión  
 $k$  : muestra k-ésima.

Debe notarse además que este enfoque empírico trata al sistema en estudio como una caja negra, de manera que no afecta la complejidad del sistema o la falta de conocimiento experto. Incluso cuando se actualicen las versiones del software bastaría con estimar nuevamente los parámetros del modelo.

En [4] se propone realizar la identificación de un sistema informático de la siguiente manera:

1. Especificar el alcance de lo que se va a modelar en base a las entradas y salidas consideradas.
2. Diseñar experimentos y recopilar datos que sean suficientes para estimar los parámetros de la ecuación diferencial lineal del orden deseado.
3. Estimar los parámetros del modelo utilizando las técnicas de mínimos cuadrados.
4. Evaluar la calidad de ajuste del modelo. Si la calidad del modelo debe mejorarse, entonces debe revisarse uno o más de los pasos anteriores.

Generalmente, apache es el servidor web elegido por ser el de mayor uso en la actualidad, alrededor del 65% del mercado de servidores web. Se elige como variable de salida  $y(k)$  al tiempo de respuesta (TR) del servidor web, ya que brinda información directa al usuario sobre la performance del servidor. En el caso de la señal de entrada  $u(k)$  del servidor web, se utiliza el valor del parámetro de configuración MaxClients (MC) que determina en todo momento el número máximo de procesos concurrentes que el servidor puede procesar.

La arquitectura sugerida para la identificación del sistema se muestra en la Figura 5. Consta de una computadora personal PC1 en la que se instala y configura el software del servidor web Apache. En la misma PC1, se debe instalar un sensor software, encargado de recoger la información de los tiempos de inicio y fin de cada http que ingresa al servidor web, información que se halla en el log del mismo servidor, para luego realizar los cálculos del tiempo de respuesta de los http completados y hallar un tiempo de respuesta promedio.

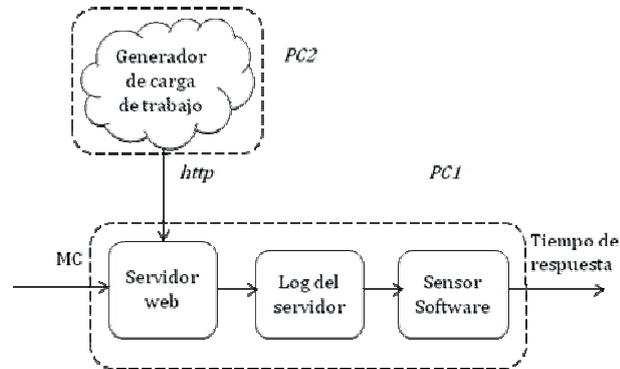


Figura 5. Arquitectura para la identificación del sistema.

En la computadora personal PC2 se instala un generador de carga de trabajo, simulador de la actividad de los usuarios que desean acceder al servidor web enviando https al servidor web. En este caso, podría ser el JMeter una aplicación software que forma parte del proyecto Apache, o cualquier que cumpla similar función. La señal de entrada MaxClients (MC) se simula de acuerdo con el rango dinámico de operación del servidor mediante una señal seno discreta para obtener una señal de salida Tiempo de Respuesta (TR) con un comportamiento similar al mostrado en la Figura 6.

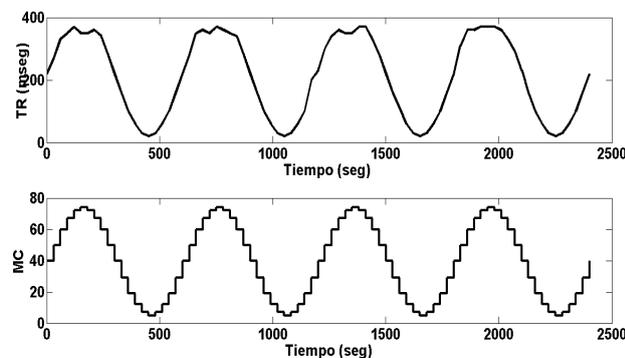


Figura 6. Señales de entrada y de salida.

Con los datos ya obtenidos mediante el sensor software, se estiman los parámetros de regresión A y B de la ecuación ARX, utilizando la técnica de mínimos cuadrados que se halla implementada en, por ejemplo, el Toolbox Identificación de Sistemas del Matlab, quien también proporciona información sobre la evaluación de la calidad de ajuste del modelo.

Finalmente, se procede al diseño y evaluación del controlador realimentado, el cual puede ser un controlador PID, o uno basado en lógica difusa, u otro basado en redes neuronales, cada uno de los cuales posee sus propias características y complejidades.

Para el diseño de controladores PID, se puede emplear el Toolbox SISOTool del Matlab que permite calcular los parámetros  $K_p$ ,  $K_I$  y  $K_D$  de la ecuación (2) del controlador:

$$u = K_p e + K_I \int e dt + K_D \frac{de}{dt} \quad (2)$$

donde  $u$  : acción de control (MC)  
 $e$  : error (TR real – TR deseado)  
 $K_p$  : parámetro proporcional  
 $K_I$  : parámetro integral  
 $K_D$  : parámetro derivativo.

Como el comportamiento de los servidores web es inherentemente no lineal, últimamente se está optando por el diseño de controladores con lógica difusa porque su performance es superior al clásico controlador PID, además que los parámetros de un controlador basado en lógica difusa se adaptan automáticamente a varios servidores, mientras que los de los controladores PID deben optimizarse para cada tipo de servidor, pero principalmente debido a que los controladores con lógica difusa, incluso con redes neuronales, no necesitan de la previa identificación del modelo del servidor web.

En el diseño de un controlador difuso que mejore el tiempo de respuesta de un servidor web, podría emplearse las reglas difusas de la Tabla 1, donde NG representa un disminución grande en el valor de MC o de TR, y PG, a un incremento grande en el valor de MC o de TR. [4]

Regla	Si		Entonces	
	Cambio actual en MC	Y	Cambio actual en TR	Próximo cambio en MC
1	NG	Y	NG	NG
2	NG	Y	PG	PG
3	PG	Y	NG	PG
4	PG	Y	PG	NG

Tabla 1. Reglas difusas para disminuir el tiempo de respuesta de un servidor web [4].

De la Tabla 1, si por ejemplo, hubo un gran incremento en el valor de MC en el tiempo  $k$ , lo que ha provocado una gran disminución en el tiempo de respuesta TR; entonces en el tiempo  $(k + 1)$  la idea es incrementar un poco más el valor de MC, lo cual expresado en una regla difusa sería similar a decir: “Si el cambio actual en el valor de MC es un incremento grande y el cambio actual en el valor de TR es una disminución grande, entonces el próximo cambio en el valor de MC debe ser un incremento grande”, como se detalla en la regla 3.

Como se puede observar, el controlador PID o el controlador difuso o cualquier otro controlador realimentado que forma parte del gestor autónomo, tratan de encontrar siempre el nuevo valor de MaxClients que minimice el tiempo de respuesta del servidor web, automáticamente. Esto es similar a decir que en la gestión autónoma, el gestor autónomo halla automáticamente el nuevo valor de MaxClients para minimizar el tiempo de respuesta del servidor web y lo actualizan de inmediato en el script de configuración del servidor sin intervención manual del administrador humano.

## 4. Conclusiones y futuros trabajos

Se observa que la teoría de control proporciona una arquitectura y un fundamento analítico sistemático para implementar controladores realimentados que forman parte importante de los sistemas de gestión autónoma.

Se observa que si existe una correspondencia entre los elementos de la arquitectura de un sistema autónomo y los elementos de un sistema de control, principalmente en lo referido a la realimentación.

La gestión autónoma junto con la teoría de control no elimina al ser humano, sino que le ayuda en la reducción de tareas rutinarias de administración de manera que pueda dedicarse mejor a la aplicación de la tecnología en el logro de los objetivos del negocio.

La teoría de control proporciona una metodología y un marco de referencia adecuado para el diseño de controladores realimentados que formen parte del gestor autónomo del elemento autónomo.

Trabajos futuros involucran el estudio de la aplicación de la teoría de control a la gestión autónoma de otro tipo de servidores y la utilización de la lógica difusa, redes neuronales y algoritmos genéticos, en el diseño de controladores automáticos.

## Referencias bibliográficas

- [1] Y. Diao, J. Hellerstein, S. Parekh, R. Griffith, G. Kaiser y D. Phung, *A Control Theory Foundation for Self-Managing Computing Systems*, IEEE DOI: 10.1109/JSAC.2005.857206, 2005.
- [2] A. Fox y D. Patterson, *Self-repairing computers*, Scientific American, Jun2003, Vol. 288, Issue 6, 2003.
- [3] IBM, *An architectural blueprint for autonomic computing*, IBM Corporation, 2006.
- [4] J. Hellerstein, Y. Diao, S. Parekh y D. Tilbury, *Feedback Control of Computing Systems*, Hoboken, New Jersey: John Wiley & Sons, Inc, 2004.
- [5] S. Keshav, *Control-theoretic approach to flow control*, ACM SIGCOMM Computer Communication Review, 1991.
- [6] C. Hollot, V. Misra, D. Towsley y W. Gong, *A control theoretic analysis of RED*, Proceedings of IEEE INFOCOM 2001, 2001.
- [7] J. Aman, C. Eilert, D. Emmes, P. Yocom y D. Dillenberger, *Adaptive algorithms for managing a distributed data processing workload*, J. Aman, C. Eilert, D. Emmes, P. Yocom, D. Dillenberger. Adaptive algorithm IBM Systems Journal, 1997.
- [8] K. Wang, M. Chen, H. Huang, W. Subramonian, C. Lu y C. Gill, *Control-Based Adaptive Middleware for Real-Time Image Transmission over Bandwidth-Constrained Networks*, IEEE Transactions on parallel and distributed systems, 2008.
- [9] N. Gandhi, M. Tilbury, J. Diao, J. Hellerstein y S. Parekh, *MIMO control of an Apache web server: modeling and controller design*, American Automatic Control Council, 2002.
- [10] S. Parekh, N. Ganchi, J. Hellerstein, D. Tilbury, T. Jayram y J. Bigus, *Using Control Theory to Achieve Service Level Objectives In Performance Management*, Kluwer Academic Publishers, Real Time Systems, 23, 127-141, 2002.
- [11] J. Hellerstein, *Challenges in Control Engineering of Computing Systems*, Proceedings of the 2004 American Control Conference, 2004.
- [12] Y. Diao, J. Hellerstein y S. Parekh, *Using fuzzy control to maximize profits in service level management*, IBM Systems Journal, Vol 41, No 3, 2002.
- [13] Y. Wei, C. Lin, T. Voigt y F. Ren, *Fuzzy Control for Guaranteeing Absolute Delays in Web Servers*, Tsinghua University, Swedish Institute of Computer Science, 2003.
- [14] R. Fontaine, P. Laurencot y A. Aussem, *Mixed Neural and Feedback Controller for Apache Web Server*, 5th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, 2009.
- [15] H. Muller, L. O'Brien, M. Klein y B. Wood, *Autonomic computing*, Carnegie Mellon University, 2006.
- [16] J. Kephart y D. Chess, *The vision of autonomic computing*, IEEE Computer Society, 2003.
- [17] F. Lewis, *Applied Optimal Control and Estimation*, Prentice Hall, 1992.
- [18] K. Ogata, *Ingeniería de control moderna*, Prentice Hall Hispanoamericana SA, 1998.
- [19] K. Astrom, *Challenges in Control Education*, Symposium on Advances in Control Education, 2006.