

# Búsqueda rápida de extremos para un sistema de primer orden

Ulises Ramírez-Carmona\* Jaime A. Moreno\*

\* Instituto de Ingeniería, Universidad Nacional Autónoma de México  
(UNAM), Coyoacán, D.F., 04510, México.  
(e-mail: URamirezC@iingen.unam.mx, JMorenoP@ii.unam.mx)

Resumen: En este artículo se presenta un algoritmo para estabilizar el punto de operación de salida óptima (punto de operación óptimo) de un sistema de primer orden, en tiempo finito. La estrategia utiliza como base un trabajo previo en el que se maximiza la producción de biogás de un biorreactor de segundo orden; en este trabajo se presenta un modelo general de un sistema de primer orden y, en el trabajo previo, se tenía un modelo de segundo orden con una estructura particular. Se trata de una ley de control conmutada con sólo dos valores de entrada, la cual conduce a las trayectorias del sistema al punto de operación óptimo, empleando únicamente la información de la salida. En contraste con las estrategias *extremum-seeking* clásicas, la estrategia de control propuesta no supone que el comportamiento de la planta en lazo cerrado es cuasi-estacionario, ni que el modelo es conocido. Por lo tanto, el algoritmo es capaz de alcanzar el punto de operación óptimo con un tiempo de convergencia del orden del tiempo de asentamiento natural del sistema, debido a que se aprovechan las características dinámicas del mismo.

*Keywords:* Optimización, control conmutado, extremum-seeking.

## 1. INTRODUCCIÓN

En el campo de control automático existen diversos tipos de problemas que pueden ser abordados, cada uno de ellos con cierta metodología o restricciones para proporcionar una solución. Por mencionar algunos: estabilización, regulación y seguimiento de trayectorias. En la mayor parte de éstos, a pesar de que los parámetros del sistema sean desconocidos, se conoce perfectamente el punto de operación en el que se desea operar a sistema o al menos se dispone de la información necesaria para calcularlo. En este sentido, el punto de operación es un dato del problema.

Otro tipo de problemas son los de búsqueda de extremos (*extremum-seeking*), en los cuales se desea identificar y estabilizar el punto de operación en que la salida es óptima (máxima/mínima). El problema general de búsqueda de extremos se puede representar de la siguiente forma.

Considere el sistema dinámico,

$$\begin{aligned} \dot{x} &= f(x, u), \\ y &= h(x), \end{aligned} \quad (1)$$

donde  $x \in \mathbb{R}^n$  es el estado,  $u \in \mathbb{R}$  es la señal de control y  $y \in \mathbb{R}$  es la salida con mapas  $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ ,  $h : \mathbb{R}^n \rightarrow \mathbb{R}$ . La variedad de puntos de operación se define como  $\{\bar{x} \in \mathbb{R}^n : f(\bar{x}, \bar{u}) = 0\}$ , y asociada a la misma existe un mapa  $\bar{u} \rightarrow \bar{x}$ . Al restringir la salida al conjunto de puntos de operación se genera un mapa estático (de estado estacionario)  $\bar{u} \rightarrow \bar{y}$  que relaciona a cada entrada constante con una salida constante.

Por otra parte, se supone que  $\bar{u} \rightarrow \bar{y}$  tiene un único

extremo absoluto, es decir, que existe un único máximo estacionario  $y^*$ . Con base en las hipótesis previas, el problema se reduce a:

- Identificar el valor de  $y^*$  en línea,
- sin requerir el conocimiento explícito de  $f(x, u)$  ni  $h(x)$ ,
- midiendo únicamente la salida  $y(t)$ .

Debido a que éste no es un problema nuevo en teoría de control, se han obtenido muchos resultados en ésta línea de investigación, con aproximaciones e ideas de solución distintas. Sin embargo, al comienzo de la exploración, muchos métodos fueron heurísticos y por ello se consideraron poco fiables.

Algunos intentos de solución provienen de la optimización numérica (método de Newton, programación lineal u optimización convexa) (Wright y Nocedal, 1999; Zhang y Ordóñez, 2011). Se supone que el problema es encontrar el extremo del mapa estático  $h(x)$  restringido a la ecuación diferencial  $\dot{x} = f(x, u)$ . Existen diversas metodologías, asumiendo que se conoce información del gradiente, o de las funciones  $f$  y  $h$ . En general, se requiere información del modelo con cierta precisión que permita implementar algún controlador de regulación que esté embebido en el método numérico.

Otro enfoque es el de sintonización robusta (Blanchini et al., 2015, 2017), los cuales resuelven el problema si  $y = f(u)$  (sistema estático), pero requieren conocer  $y^*$  y las propiedades de  $\frac{\partial y}{\partial u}$ .

También se ha resuelto el problema con modos deslizantes, nuevamente cuando el sistema es estático  $y = f(u)$

(Korovin y Utkin, 1974; Utkin, 2013) al agregar una dinámica que permite realizar la optimización. Ello se realiza con sistema estático escalar ( $u \in \mathbb{R}$ ) y se plantea la extensión al caso en que  $u \in \mathbb{R}^n$  seleccionando direcciones búsqueda de manera similar a como lo hacen los métodos numéricos directos.

Finalmente, las técnicas conocidas como *extremum-seeking* permiten lograr la solución si el sistema es estático  $y = f(u)$ , o si el estado se mueve lentamente, es decir,  $\dot{x} \approx 0$  (Krstić y Wang, 2000; Ariyur y Krstić, 2003; Liu y Krstić, 2012). O combinando control de aprendizaje iterativo con *extremum-seeking* para mapas cíclicos Cao et al. (2017).

Hay más propuestas de solución, y como se puede apreciar, el problema es complejo y ninguna propuesta contempla la solución completa del mismo, puesto que si se desconoce el modelo, es necesario asumir que el sistema no tiene dinámica y si se supone que el sistema tiene dinámica, se tiene que poseer información del modelo.

Sin embargo el problema que se desea resolver aquí es más modesto, se plantea la solución para un sistema de primer orden en el que se considere la dinámica del sistema y desconociendo el modelo matemático del mismo.

## 2. PLANTEAMIENTO DEL PROBLEMA

Considere un sistema dinámico descrito por la siguiente ecuación diferencial de primer orden

$$\begin{aligned}\dot{x} &= f(x, u), \\ y &= h(x),\end{aligned}\quad (2)$$

donde  $x \in \mathbb{R}$  es el estado,  $u \in \mathbb{R}$  es la señal de control y  $y \in \mathbb{R}$  es la salida. Se supone que la señal de control está saturada  $u \in [u_1, u_2] \subset \mathbb{R}$ , y que cuando  $u$  varía en ese compacto, tomando valores constantes  $\bar{u}$ , se genera una variedad de puntos de equilibrio de la siguiente manera:

$$X = \{\bar{x} \in \mathbb{R} | f(\bar{x}, \bar{u}) = 0, \bar{u} \in [u_1, u_2]\}. \quad (3)$$

Se supone que cuando  $u = u_i$ ,  $i = 1, 2$ , el sistema (2) tiene un punto de equilibrio  $x_i$  asintóticamente estable, y que el mapa de la salida en equilibrio  $y = h(\bar{x})$  alcanza un máximo en  $x = x^*$  ( $u = u^*$ ), es decir, se cumple que:

$$\left. \frac{dh(\bar{x}(\bar{u}))}{d\bar{u}} \right|_{\bar{u}=u^*} = 0, \quad \left. \frac{d^2h(\bar{x}(\bar{u}))}{d\bar{u}^2} \right|_{\bar{u}=u^*} < 0. \quad (4)$$

El problema consiste en diseñar un algoritmo de control que maximice el valor de  $y$  en estado estacionario, pero sin requerir el conocimiento de  $u^*$  ni de las funciones  $f$  y  $h$ . En caso de que se desee minimizar  $y$ , el problema se resuelve de la misma forma al reemplazar  $y$  por  $-y$  en el diseño. Adicionalmente, se supone que en la región donde se desea realizar la búsqueda el extremo es único y que la función es al menos dos veces continuamente diferenciable.

## 3. ALGORITMO DE CONTROL

La estrategia propuesta tiene como sustento teórico los sistemas con puntos de equilibrio conmutados y está basado en el algoritmo propuesto por Ramírez-Carmona

et al. (2018). La idea básica consiste en conmutar la entrada  $u$  entre dos valores posibles  $u \in \{u_1, u_2\}$  (las cotas de la saturación), de forma que con cada valor de entrada se tenga un punto de operación distinto  $x_1$  y  $x_2$ . La señal de control conmuta cada que la salida  $y(t)$  decrece o permanece constante, dicho de otra forma,  $y(t)$  conmuta cada que  $\dot{y}(t) \leq 0$ . Lo anterior está sintetizado en el Algoritmo 1, donde  $t_k$  es el  $k$ -ésimo instante de muestreo, y  $t_k^-$  es un valor de tiempo arbitrariamente cercano a  $t_k$  por la izquierda. Debido a que para la implementación sólo se necesita el signo de la derivada de  $y(t)$  con respecto al tiempo, éste puede aproximarse como  $\text{sign}(\dot{y}(t)) \approx \text{sign}(y(t_k) - y(t_{k-1}))$ . Al inicio puede suponerse que  $u(t_0) = u_1$ .

---

### Algorithm 1 Ley de actualización de $u$ , $k = 1, 2, 3, \dots$

---

**Data:**  $\dot{y}(t_k^-)$ ,  $u(t_k^-)$   
**if**  $\text{sign}(\dot{y}(t_k^-)) \leq 0$  **then**  
    **if**  $u(t_k^-) == u_1$  **then**  
         $u(t_k) = u_2$   
    **else**  
         $u(t_k) = u_1$   
    **end**  
**else**  
     $u(t_k) = u(t_k^-)$   
**end**

---

Como se puede apreciar, el algoritmo intenta que la derivada de  $y(t)$  con respecto al tiempo sea positiva en todo instante, o equivalentemente que  $y(t)$  siempre crezca. No obstante, como  $x_1$  es un punto de operación asintóticamente estable, existen condiciones en las cuales  $y(t)$  podría crecer, sin que exista una conmutación de la señal de control. Esto ocurre cuando la condición inicial está fuera de la variedad de puntos de operación  $x(t_0) = x_0 \notin X$  y el estado está más cerca del punto de operación que corresponde a la entrada  $u_1$ , es decir,  $|x_0 - x_1| < |x_0 - x_2|$ .

Una forma de arreglar el problema es modificar ligeramente el algoritmo, añadiendo una conmutación forzada en caso de que, después de determinado número de tiempos de muestreo, no haya habido conmutación. Con ello se obtiene el Algoritmo 2. Las modificaciones muestran en un color distinto, para que sean más fáciles de visualizar.

En esta nueva versión del algoritmo de control, se define una variable auxiliar  $j$  con un valor inicial de cero. El valor de 100 es un ejemplo, y podría ser más grande o más pequeño sin modificar apreciablemente el desempeño del sistema en lazo cerrado. De hecho, como se mencionó, es una forma de asegurar la convergencia global al punto de equilibrio óptimo  $x^*$ . Nuevamente, para implementación, el signo de  $\dot{y}(t)$  puede aproximarse con  $\text{sign}(y(t_k) - y(t_{k-1}))$ , y al inicio puede suponerse que  $u(t_0) = u_1$ .

## 4. EL SISTEMA ESCALAR VISTO COMO SISTEMA CONMUTADO

El sistema (2) analizado cuando la entrada sólo puede tomar los valores  $u_1$  y  $u_2$  es equivalente al sistema conmutado (5).

---

**Algorithm 2** Ley de actualización de  $u$ ,  $k = 1, 2, 3, \dots$ 


---

**Data:**  $\dot{y}(t_k^-)$ ,  $u(t_k^-)$

```

if  $\text{sign}(\dot{y}(t_k^-)) \leq 0$  then
  if  $u(t_k^-) == u_1$  then
     $u(t_k) = u_2$ 
  else
     $u(t_k) = u_1$ 
  end
   $j = j + 1$ 
else
   $u(t_k) = u(t_k^-)$ 
   $j = j + 1$ 
end
if  $j \geq 100$  then
  if  $u(t_k^-) == u_1$  then
     $u(t_k) = u_2$ 
  else
     $u(t_k) = u_1$ 
  end
   $j = 0$ 
end

```

---

$$\begin{aligned} \dot{x} &= f(x, u_\sigma), \\ y &= h(x). \end{aligned} \quad (5)$$

donde  $f(x_i, u_i) = 0$  ( $i = 1, 2$ ) y  $\sigma \equiv \sigma(t, x)$  ( $\sigma \in \{1, 2\}$ ) es una ley de conmutación continua a tramos por la derecha que determina cuál de los 2 puntos de equilibrio  $x_1, x_2$  está activo. A continuación, se definirán algunos conjuntos que serán útiles más tarde. Primero, dos intervalos de puntos de operación  $X_1, X_2 \subset X$ , los cuales corresponden a los puntos de operación que se encuentran entre  $x_1$  y  $x^*$ , y entre  $x_2$  y  $x^*$ , respectivamente.

$$X_1 = \{\bar{x} \in X | \bar{u} \in (u_1, u^*)\}, \quad (7)$$

$$X_2 = \{\bar{x} \in X | \bar{u} \in (u^*, u_2)\}. \quad (8)$$

Sin pérdida de generalidad, se supone que  $x_1 < x_2$  (la argumentación para el otro caso es la misma, simplemente tendrían que intercambiarse los valores de  $u_1$  y  $u_2$ , de forma que  $x_1 < x_2$ ). En la Figura 1 se muestran los puntos de equilibrio, con la forma del campo vectorial en cada caso.

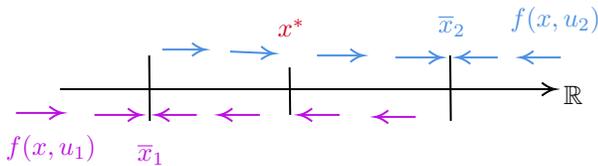


Figura 1. Espacio de estados, con los campos vectoriales.

## 5. PRUEBA DE ESTABILIDAD

Para verificar que el algoritmo propuesto resuelve el problema, es necesario resumir las condiciones bajo las cuales funciona y probar que dicho resultado es correcto. Lo anterior se realiza en el Teorema 1:

**Teorema 1.** Suponga que el mapa  $h(x)$  del sistema (2) tiene un extremo absoluto en  $x^* \in X$ , y que cuando la entrada es constante y toma el valor  $u = u_i$  ( $i = 1, 2$ ), el

sistema (2) posee un punto de equilibrio asintóticamente estable  $x_i$ . Entonces, el Algoritmo 2 implementado en tiempo continuo estabiliza  $x^*$  en tiempo finito.

**Demostración.** Dado que  $x_\sigma$  es un punto de operación asintóticamente estable para  $\sigma(t) = i$ , existe una función de Lyapunov  $V(x - x_i) = \frac{1}{2}(x - x_i)^2$ , que prueba estabilidad de  $x_i$  para  $\dot{x} = f(x, u_i)$ . En este caso, para el sistema (5), se propone la siguiente función candidata de Lyapunov:

$$V(x - x^*) = \frac{1}{2}(x - x^*)^2, \quad (9)$$

la cual es  $V(0) = 0$  ( $x = x^*$ ), y  $\lim_{|x| \rightarrow \infty} V(x - x^*) = \infty$ , es decir, radialmente no acotada. Tomando la derivada de  $V(x - x^*)$  a lo largo de las trayectorias de (5), se obtiene:

$$\dot{V}(x - x^*) = (x - x^*)f(x, u_\sigma). \quad (10)$$

Independientemente de si la entrada es  $u_1$  o  $u_2$ , el campo vectorial siempre apunta en dirección del conjunto  $X$ . Además, la conmutación forzada evita que las trayectorias se atasquen en  $x_1$  o  $x_2$ , lo cual asegura que después de un tiempo  $t_c$  el estado se encontrará en el conjunto  $X$ . Como la función  $h(x)$  tiene un extremo absoluto en  $x^*$ , su comportamiento cualitativo es como el que se muestra en la Figura 2.

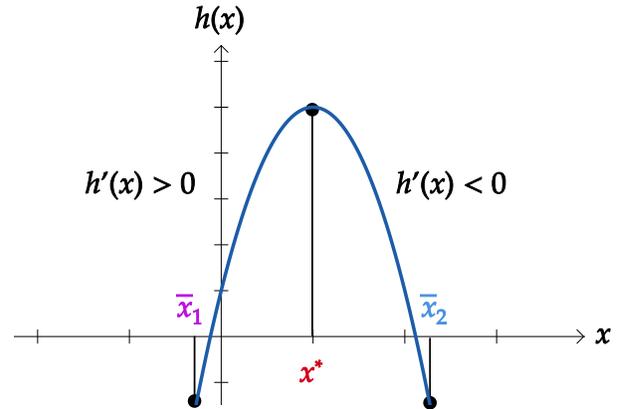


Figura 2. Función  $h(x)$  restringida a  $X$ .

Al analizar los casos posibles de la derivada de  $\dot{V}(x - x^*)$ , cuando el estado se encuentra en  $X$ , se tiene lo siguiente:

- Si  $x(t) \in X_1$ , entonces  $(x - x^*) < 0$ , debido a Algoritmo 2 el control que permanece activo en dicha región es  $u_2$ , con su campo vectorial asociado  $f(x, u_\sigma) = f(x, u_2) > 0$ , lo que implica que su producto es negativo.

$$\dot{V}(x - x^*) < 0. \quad (11)$$

- Si  $x(t) \in X_2$ , entonces  $(x - x^*) > 0$ , debido a Algoritmo 2 el control que permanece activo en dicha región es  $u_1$ , con su campo vectorial asociado  $f(x, u_\sigma) = f(x, u_1) < 0$ , lo que implica que su producto es negativo.

$$\dot{V}(x - x^*) < 0. \quad (12)$$

- Si  $x(t) = x^*$ , entonces la derivada de la función candidata de Lyapunov es nula.

$$\dot{V}(x^* - x^*) = 0. \quad (13)$$

Lo anterior implica que  $x^*$  es un punto de equilibrio asintóticamente estable del sistema (2). Ahora, para probar que la convergencia es en tiempo finito, sólo es necesario argumentar que cuando el estado se encuentra en el conjunto  $X_1$  ( $X_2$ ), el sistema activo es  $\sigma(t) = 1$  ( $\sigma = 2$ ),  $x_i$  es asintóticamente estable y por ello tendría que pasar forzosamente por  $x^*$  para poder llegar a  $x_i$ . ■

## 6. EJEMPLOS DE SIMULACIÓN

Se considera el sistema de primer orden

$$\begin{aligned} \dot{x} &= -x + u, \\ y &= -(x - 2)^2 + 5, \end{aligned} \quad (14)$$

donde  $x \in \mathbb{R}$  es el estado,  $y \in \mathbb{R}$  es la salida. El punto crítico es  $x^* = 2$  y los puntos de operación cumplen la relación  $\bar{x} = \bar{u}$ . Para que el problema posea solución, las cotas de la saturación deben de cumplir la desigualdad  $u_1 < x^* < u_2$ . Para este ejemplo se escogieron ( $u_1 = -1$  y  $u_2 = 5$ ). Si se toman como condiciones iniciales  $x(0) = -3$  y  $u(0) = u_1$ , se obtiene el comportamiento que se muestra en la Figura 3, en donde se puede observar que el extremo de la función  $h(x)$  se identificó correctamente.

Para realizar la simulación se utilizaron dos tiempos de muestreo, el primero es el paso de integración ( $1 \times 10^{-5}$ ) y el segundo corresponde al tiempo de decisión ( $1 \times 10^{-3}$ ), es decir, el tiempo de muestreo del controlador. Lo anterior, con la finalidad de tener un sistema continuo con una implementación discreta del controlador.

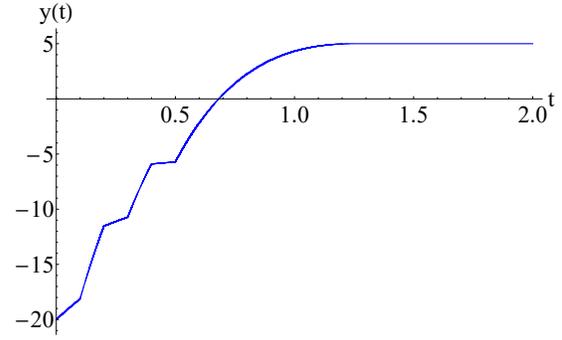
El ejemplo muestra que el punto de equilibrio  $\bar{x} = x^*$  se estabiliza tiempo finito, a pesar de que originalmente  $x^*$  no era un punto de equilibrio de los sistemas  $f(x, u_1)$  y  $f(x, u_2)$ . Como se puede apreciar en las Figuras 3a, 3b, y 3c la conmutación forzada ocurre cuando  $x(t) < -1$ , lo cual permite que la trayectoria no se atasque en el punto de equilibrio  $x_1 = -1$ . Una vez que el estado ingresa en el conjunto  $X = \{x \in \mathbb{R} | -1 \leq x \leq 5\}$ , la conmutación forzada sigue existiendo. Sin embargo, ésta no afecta el desempeño del lazo cerrado porque en el siguiente instante, el algoritmo detecta que la entrada aplicada es incorrecta y realiza la modificación pertinente del valor de  $u$ .

Finalmente, cabe mencionar que el Algoritmo 1 es capaz de determinar el extremo del sistema sin dinámica  $y = h(u)$  (con entrada escalar). Para conseguirlo se puede utilizar el sistema dinámico auxiliar:

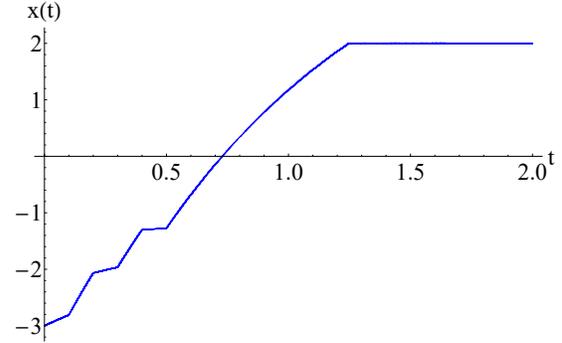
$$\dot{x} = u, \quad (15)$$

$$y = h(x) \quad (16)$$

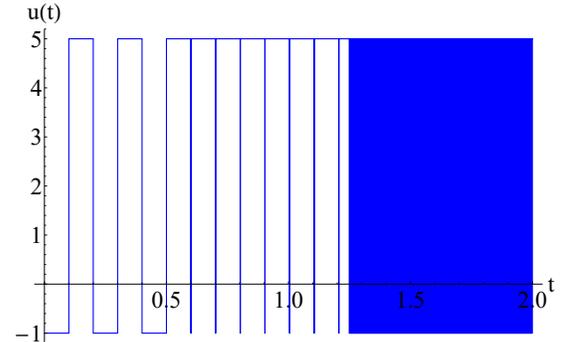
donde  $x \in \mathbb{R}$ ,  $u \in \mathbb{R}$ , con  $u_1 < 0 < u_2$ . Este procedimiento es una alternativa (cuando la entrada es escalar) al algoritmo de control presentado por Korovin y Utkin (1974).



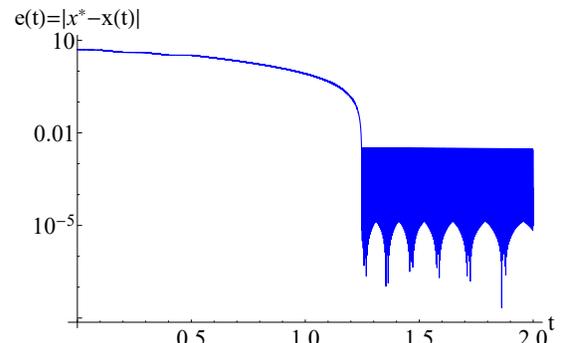
(a) Salida  $y(t)$ .



(b) Estado  $x(t)$ .



(c) Entrada  $u(t)$ .



(d) Gráfica logarítmica del error  $e(t) = |x^* - x(t)|$ .

Figura 3. Búsqueda del extremo de  $h(x) = -(x - 2)^2 + 5$ .

## 7. CONCLUSIONES

Se presentó un algoritmo de control que permite identificar y estabilizar el punto de operación óptimo de un sistema de primer orden, en tiempo finito. Su implementación no requiere del conocimiento del modelo del sistema,

pero aprovecha su comportamiento dinámico para lograr la optimización.

Las simulaciones muestran que si el extremo es único en el intervalo de interés (función unimodal en  $u \in [u_1, u_2]$ ), se logra estabilidad global del punto de operación óptimo.

Este algoritmo mostró ser válido para el modelo de primer orden y en el trabajo previo se demostró que funciona para maximizar la producción de biogás de un biorreactor de segundo orden. Por ello, como trabajo a futuro se plantea su extensión a una clase de sistemas de segundo orden y posteriormente a sistemas de orden superior.

## AGRADECIMIENTOS

Este trabajo fue apoyado por PAPIIT-UNAM, proyectos IN113614 y 104016; Fondo de Colaboración II-FI UNAM, Proyecto IISGBAS-100-2015; CONACyT, proyecto 241171; y CONACyT CVU 780054.

## REFERENCIAS

- Ariyur, K.B. y Krstić, M. (2003). *Real-time optimization by extremum-seeking control*. John Wiley & Sons.
- Blanchini, F., Fenu, G., Giordano, G., y Pellegrino, F.A. (2015). Plant tuning: a robust lyapunov approach. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 1142–1147. IEEE.
- Blanchini, F., Fenu, G., Giordano, G., y Pellegrino, F.A. (2017). Discrete-time trials for tuning without a model. *IFAC-PapersOnLine*, 50(1), 1539–1544.
- Cao, Z., Dürr, H.B., Ebenbauer, C., Allgöwer, F., y Gao, F. (2017). Iterative learning and extremum seeking for repetitive time-varying mappings. *IEEE Transactions on Automatic Control*, 62(7), 3339–3353.
- Korovin, S. y Utkin, V. (1974). Using sliding modes in static optimization and nonlinear programming. *Automatica*, 10(5), 525–532.
- Krstić, M. y Wang, H.H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4), 595–601.
- Liu, S.J. y Krstić, M. (2012). *Stochastic averaging and stochastic extremum seeking*. Springer Science & Business Media.
- Ramírez-Carmona, U., Moreno, J.A., y Vargas, A. (2018). Fast extremum seeking for bioreactors using a variable structure control approach. In *2018 15th International Workshop on Variable Structure Systems (VSS)*, 337–342. IEEE.
- Utkin, V.I. (2013). *Sliding modes in control and optimization*. Springer Science & Business Media.
- Wright, S. y Nocedal, J. (1999). Numerical optimization. *Springer Science*, 35(67-68), 7.
- Zhang, C. y Ordóñez, R. (2011). *Extremum-seeking control and applications: a numerical optimization-based approach*. Springer Science & Business Media.