

Sintonización De Un Controlador Pid Para Un Péndulo Invertido Mediante Algoritmos Meta-Heurísticos: Luciérnaga Y Recocido Simulado

Pid Controller Tuning For An Inverted Pendulum Through Meta-Heuristic Algorithms: Firefly And Simulated Annealing

Zocimo Ñaupari ¹0000-0002-8372-7760

Luis Beltrán ^{1*}0000-0002-8372-7760

¹ Universidad Nacional de Ingeniería, Facultad de Ingeniería Eléctrica y Electrónica, Lima, Perú.

***Autor de la correspondencia:** Luis Beltrán, E-mail: lbeltranc@uni.pe

RESUMEN

En las plantas industriales un gran porcentaje de los controladores está basado en el Algoritmo de Control PID, esto por su sencillez y robustez, se tiene conocimiento que la performance de la Planta se puede ver afectada en cantidad y calidad del producto debido a la inadecuada determinación de los parámetros del controlador, incluso perjudicar a los mismos actuadores. En este trabajo, se muestra una alternativa a los métodos clásicos de sintonía de dichos parámetros: los algoritmos metaheurísticos, los cuales pertenecen a la rama de Búsqueda Informada dentro de la Inteligencia Artificial, cuyo objetivo es optimizar una determinada función de costo evitando los máximos o mínimos locales. Se consideró el modelo linealizado (para fines prácticos) de un sistema de péndulo invertido con un carro deslizante para la aplicación de dos algoritmos: el recocido simulado y el de luciérnagas; los cuales tuvieron como rango superior el determinado por el Criterio de Routh-Hurwitz para todas las especificaciones de rendimiento en el dominio del tiempo. En los resultados se pudo determinar que los parámetros optimizados del controlador PID no tenían una diferencia significativa y la velocidad de convergencia era rápida, esto nos permitiría concluir que es una opción adicional para la sintonía de lazos de control.

Palabras Clave: algoritmo, luciérnaga, meta-heurísticos, recocido simulado, controlador PID, péndulo invertido, sintonía, modelamiento, función de transferencia, índice ITSE.

ABSTRACT

In industrial plants a large percentage of the controllers is based on the PID Control Algorithm, due to its simplicity and robustness, it is known that the performance of the Plant may be affected in quantity and quality of the product due to inadequate determination of the

controller parameters, even harming the actuators themselves. In this work, an alternative to classical methods of tuning of these parameters is shown: the metaheuristic algorithms, which belong to the branch of Informed Search within Artificial Intelligence, whose objective is to optimize a certain cost function avoiding maximums or local minimums. It has considered the linearized model (for practical purposes) of an inverted pendulum system with a sliding carriage for the application of two algorithms: simulated annealing and fireflies; which had as a higher rank the one determined by the Routh-Hurwitz Criterion for all the specifications of time domain performance. In the results it was possible to determine that the optimized parameters of the PID controller did not have a significant difference and the speed of convergence was fast, this would allow us to conclude that it is an additional option for the tuning of control loops.

Keywords: *firefly, meta-heuristic, simulated annealing, algorithm, PID controller, inverted pendulum, tuning, modelling, transfer function, ITSE index*

1. INTRODUCCIÓN

El péndulo invertido es un sistema que consta de un péndulo que es libre de oscilar alrededor de un punto pivote (para un mismo plano) con el objetivo de mantener el péndulo en posición vertical. El punto pivote está sobre un carro que ofrece el desplazamiento horizontal, el cual permitirá balancear; para una perturbación pequeña expresada en grados; el sistema en mención debido a su alta inestabilidad. Este sistema es importante por sus aplicaciones prácticas donde se requiera balance como en acciones que involucre caminar.

Se han realizado varios intentos con diferentes técnicas para lograr el equilibrio del péndulo, en [1] una red neuronal es entrenada para generar acciones sobre un péndulo invertido, en [2] se hace uso del algoritmo luciérnaga para determinar los parámetros de un controlador PID y así mantener el equilibrio del péndulo. En [3] se implementó experimentalmente un controlador fuzzy que usa visión feedback, lo cual busca adquirir los datos del sistema a través del procesamiento de imágenes y no de sensores, el objetivo fue explorar las condiciones críticas del controlador y su velocidad de respuesta.

Para corregir el error definido como la diferencia del ángulo del péndulo respecto a la vertical (el cual debería ser igual a cero) vamos a considerar un controlador PID, los cuales han ganado una popularidad atribuida parcialmente a su comportamiento robusto en un rango amplio de condiciones de operación y parcialmente también a su simplicidad funcional que permite a los ingenieros operarlos de una forma simple y directa [4], para definir los parámetros de dicho controlador (k_p , k_i , k_d) en la literatura se ha propuesto muchos tipos diferentes de reglas de sintonización, que permiten llevar a cabo una sintonización delicada y fina [5], sin embargo el problema de determinar los parámetros es en realidad un problema de optimización, donde en términos de la variable del sistema, se hallen los parámetros que minimicen un criterio de error [6]

2. MODELAMIENTO DEL PÉNDULO INVERTIDO

En la figura 1 se muestra el péndulo invertido considerado en este artículo, en [7] se tiene los parámetros utilizados.

El péndulo tiene su CG (centro de gravedad) en el centro de la varilla y se considera que sólo se puede mover en un plano solamente (xy). La ecuación que describe el movimiento horizontal del carro es:

$$f(t) - b\dot{x} - N = M\ddot{x}$$

$$f(t) = M\ddot{x} + b\dot{x} + N \quad (1)$$

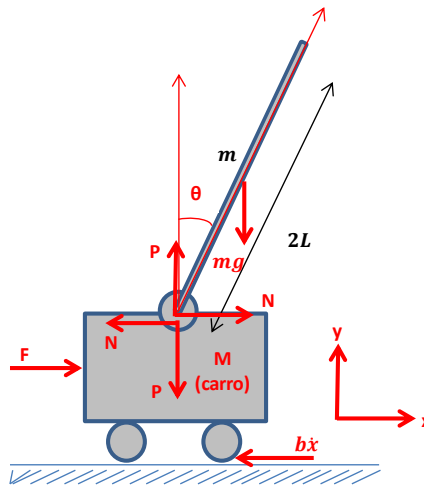


Fig. 1. Diagrama de Péndulo Invertido

Donde $f(t)$ es la fuerza aplicada en el carro, $b\dot{x}$ es la fuerza de recuperación debido a la fricción, N y P son las fuerzas de reacción del péndulo en los ejes x e y respectivamente, M es la masa del carro, m es la masa de la varilla, $M\ddot{x}$ es la fuerza inercial debido a la masa del carro. Como se comentó el movimiento vertical del carro no se asumirá.

Aplicando la segunda ley de Newton en el eje x para la varilla del péndulo, el CG del péndulo se ubica en el punto medio de la varilla:

$$N = m\ddot{x} + mL\ddot{\theta}\cos\theta - mL\dot{\theta}^2 \sin\theta \quad (2)$$

Donde θ es el ángulo recorrido alrededor del punto pivote del péndulo, $\dot{\theta}$ es la velocidad angular, $\ddot{\theta}$ es la aceleración angular y L es la longitud de la varilla.

Reemplazando (2) en la ecuación (1),

$$f(t) = (M + m)\ddot{x} + b\dot{x} + mL\ddot{\theta}\cos\theta - mL\dot{\theta}^2 \sin\theta \quad (3)$$

Aplicando la 2da Ley de Newton para las fuerzas perpendiculares a la varilla del péndulo:

$$P \sin \theta - mg \sin \theta - N \cos \theta = -mL\ddot{\theta} - m\dot{x} \cos \theta \quad (4)$$

Donde g es la aceleración de la gravedad.

Aplicando la 2da Ley de Newton para el movimiento de rotación de la varilla del péndulo sobre su masa:

$$LP \sin \theta - LN \cos \theta = I \ddot{\theta} \quad (5)$$

Donde I es el momento de inercia de la varilla.

Reemplazando (5) en la ecuación (4),

$$(I + mL^2)\ddot{\theta} - mgL \sin \theta + mL\dot{x} \cos \theta = 0 \quad (6)$$

Para representar las ecuaciones en el espacio de estados se considera las siguientes variables de estado:

$$\begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} \quad (7)$$

Así se obtiene:

$$\begin{bmatrix} \dot{x}1 \\ \dot{x}2 \\ \dot{x}3 \\ \dot{x}4 \end{bmatrix} = \begin{bmatrix} x2 \\ \frac{((M+m)mgL(\sin x1) + mLb(x4)(\cos x1) - (mL)^2(x2)^2(\sin x1) - mLf(\cos x1))}{((I+mL^2)(M+m) - (mL)^2(\cos x1)^2)} \\ x4 \\ -(I + mL^2) \left[\frac{(M+m)g(\sin x1) + b(x4)(\cos x1) - mL(x2)^2(\sin x1) - F(\cos x1)}{(\cos x1)((I+mL^2)(M+m) - (mL)^2(\cos x1)^2)} \right] + g(\tan x1) \end{bmatrix} \quad (8)$$

Dado que las salidas son el desplazamiento angular y el desplazamiento del carro, la ecuación de estado de salida será:

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} \quad (9)$$

A. Linealización del Modelo

Para la aplicación del controlador PID al modelo de la ecuación (8) y para fines de este trabajo se requiere tener el modelo lineal del péndulo invertido. Dado que el objetivo es que el péndulo se mantenga en posición vertical ($\theta \approx 0^\circ$) y que el controlador PID actúe como regulador ante las pequeñas perturbaciones que puedan presentarse se considera que el punto de operación (este punto deberá cumplir con las condiciones de estado de equilibrio) será: $x = [0 \ 0 \ 0 \ 0]$ y $F = [0]$.

Considerando la metodología de [8], se aproximará el modelo lineal mediante la expansión en Series de Taylor despreciando los términos de orden superior.

De lo visto líneas arriba se define:

$$\dot{x} = f(x, F)$$

$$f = [f_1 \ f_2 \ f_3 \ f_4]^T = [\dot{x}_1 \ \dot{x}_2 \ \dot{x}_3 \ \dot{x}_4]^T \quad (10)$$

En donde F es la señal de control y a partir de ahora se denotará por u y \dot{x} es el vector de estados definido anteriormente.

La expansión de las Series de Taylor de (10) sería dado por:

$$\dot{x} = f(0,0) + \frac{\partial f(0,0)}{\partial x} x + \frac{\partial f(0,0)}{\partial u} u \quad (11)$$

Resolviendo las operaciones jacobianas, (11) queda así:

$$\dot{x} = f(0,0) + \begin{bmatrix} \frac{\partial f_1(0,0)}{\partial x_1} & \frac{\partial f_1(0,0)}{\partial x_2} & \frac{\partial f_1(0,0)}{\partial x_3} & \frac{\partial f_1(0,0)}{\partial x_4} \\ \frac{\partial f_2(0,0)}{\partial x_1} & \frac{\partial f_2(0,0)}{\partial x_2} & \frac{\partial f_2(0,0)}{\partial x_3} & \frac{\partial f_2(0,0)}{\partial x_4} \\ \frac{\partial f_3(0,0)}{\partial x_1} & \frac{\partial f_3(0,0)}{\partial x_2} & \frac{\partial f_3(0,0)}{\partial x_3} & \frac{\partial f_3(0,0)}{\partial x_4} \\ \frac{\partial f_4(0,0)}{\partial x_1} & \frac{\partial f_4(0,0)}{\partial x_2} & \frac{\partial f_4(0,0)}{\partial x_3} & \frac{\partial f_4(0,0)}{\partial x_4} \end{bmatrix} * x + \begin{bmatrix} \frac{\partial f_1(0,0)}{\partial F} \\ \frac{\partial f_2(0,0)}{\partial F} \\ \frac{\partial f_3(0,0)}{\partial F} \\ \frac{\partial f_4(0,0)}{\partial F} \end{bmatrix} * u \quad (12)$$

Aplicando a la ecuación (8) se obtiene finalmente el modelo lineal:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mLg(M+m)}{I(M+m) + MmL^2} & 0 & 0 & \frac{mLb}{I(M+m) + MmL^2} \\ 0 & 0 & 1 & 0 \\ \frac{-(mL)^2g}{I(M+m) + MmL^2} & 0 & \frac{-(I + mL^2)b}{I(M+m) + MmL^2} & 0 \end{bmatrix} * x + \begin{bmatrix} 0 \\ \frac{-mL}{I(M+m) + MmL^2} \\ 0 \\ \frac{(I + mL^2)}{I(M+m) + MmL^2} \end{bmatrix} * u$$

(13)

B. Función de Transferencia de la Posición Angular y Controlador PID

Resolviendo la ecuación de estado (13) y reordenando se obtiene la función de transferencia de la posición angular.

$$\frac{\theta(s)}{U(s)} = \frac{-mLs}{\left\{ \begin{aligned} &[(I + mL^2)(M + m) - (mL)^2]s^3 + [(I + mL^2)b]s^2 \\ &- [mgL(M + m)]s - [mgLb] \end{aligned} \right\}} \quad (13)$$

Reemplazando los valores según [8] se obtiene:

$$\frac{\theta(s)}{U(s)} = \frac{-s}{0.2201s^3 + 0.04002s^2 - 6.863s - 0.9804} \quad (14)$$

No se va a tomar en cuenta la función de transferencia de la posición del carro, dado que no es el objetivo realizar control sobre este, sino es regular la posición vertical (siempre el set point será: $\theta = 0^\circ$) ante las perturbaciones que se presentan y para ello la acción de control: F (fuerza) será ejercida sobre el carro, de hecho el mismo controlador PID afectaría ambas funciones de transferencia dado que no son independientes, en la ejecución se tendría que tener una mesa larga para dar disponibilidad a que el carro pueda recorrerlo sin limitación.

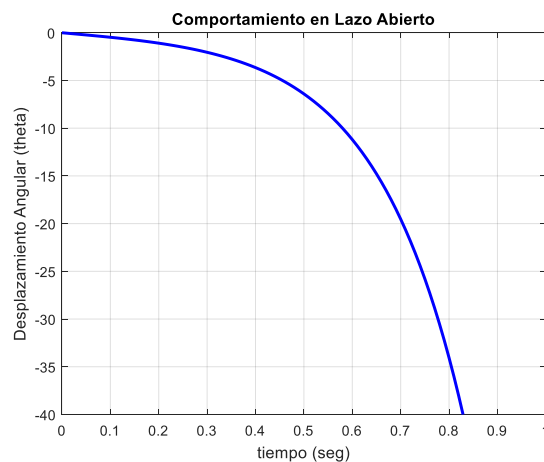


Fig. 2. Respuesta al impulso en lazo abierto para θ

Como se aprecia en la figura 2, se observa que el sistema no es estable en la condición de lazo abierto, para una perturbación simulada mediante la función impulso. Esto también es verificado en (14) dado que tenemos en el denominador coeficientes negativos, mas estrictamente tenemos un polo positivo (lado derecho del plano s complejo).

Debido a lo anterior se requiere diseñar algún tipo de control para mejorar la respuesta, en esta oportunidad se aplicará un controlador de tipo PID (proporcional-integral-derivativo) que es muy común y fácil de implementar.

El diagrama de bloques se presenta en la figura 3 y la función de transferencia de un controlador PID es dado por:

$$PID(s) = \left(k_p + \frac{k_i}{s} + k_d s \right) \quad (15)$$

Donde: k_p es la ganancia proporcional, k_i es la ganancia integral y k_d es la ganancia derivativa.

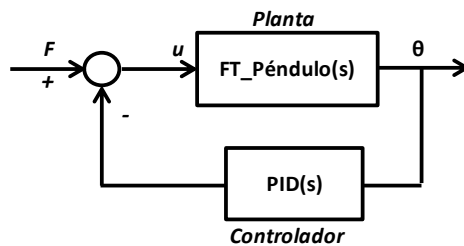


Fig. 3. Diagrama de bloques del sistema realimentado con el controlador PID

La función de transferencia de lazo cerrado del sistema puede ser escrito como:

$$\frac{\theta(s)}{F(s)} = T(s) = \frac{FT_Péndulo(s)}{1 + PID(s) * FT_Péndulo(s)} \quad (16)$$

Resolviendo se obtiene:

$$T(s) = \frac{-s}{\left[\begin{array}{l} 0.2201s^3 + (0.04002 - k_d)s^2 - (6.863 + k_p)s - \dots \\ \dots - (0.9804 + k_i) \end{array} \right]} \quad (17)$$

La ecuación característica obtenida de la ecuación (17) es:

$$0.2201s^3 + (0.04002 - k_d)s^2 - (6.863 + k_p)s - \dots - (0.9804 + k_i) = 0 \quad (18)$$

Para la estabilidad los coeficientes de la ecuación característica deben ser positivos. Por lo tanto, se tiene los siguientes rangos:

$$k_p < -6.863, k_1 < -0.9804, k_d < 0.04002 \quad (19)$$

3. ALGORITMOS METAHERÚSTICOS

Los algoritmos metaheurísticos son una nueva perspectiva con la que se afronta diferentes tipos de problemas: estrategia-resultados [9], bajo esta premisa el algoritmo heurístico busca soluciones lo suficientemente buenas, sin ser necesariamente la mejor o la óptima, estos buscan reducir el esfuerzo computacional de tiempo y eficiencia a costa de la precisión para llegar a la solución.

Blum y Roli [10] consideran 9 propiedades de la metaheurística:

- Son estrategias que guían el proceso de búsqueda.
- Exploran eficientemente el espacio de búsqueda para encontrar soluciones (casi) óptimas.
- Las técnicas que lo constituyen van desde simples procedimientos de búsqueda local hasta complejos procesos de aprendizaje.
- Son aproximados y generalmente no deterministas.
- Pueden incorporar mecanismos para evitar quedar atrapados en áreas confinadas del espacio de búsqueda. Esto les permite poder “escapar” de puntos críticos locales lo cual permite tener una mayor probabilidad de alcanzar puntos críticos globales.
- Los conceptos básicos de la metaheurística permiten una descripción por niveles abstracta.
- No son específicas para un determinado problema.
- Pueden hacer uso del conocimiento específico de un dominio en forma de heurísticas que son controladas por la estrategia de nivel superior.
- Las metaheurísticas más avanzadas de hoy utilizan la experiencia de búsqueda (embebida en alguna forma de memoria) para guiar la búsqueda.

Se tiene una clase amplia de algoritmos metaheurísticos, los que en esta oportunidad se discutirán son los inspirados en el comportamiento de sistemas de la naturaleza: algoritmo luciérnaga y recocido simulado. Para mayor detalle se puede consultar a [9], [10], [11], [12].

A. Algoritmo Luciérnaga

Este algoritmo fue desarrollado por Xin-She Yang en el 2007 (Universidad de Cambridge). Para su implementación se debería considerar 3 reglas idealizadas [12], [13]:

1. Todas las luciérnagas son unisex así que una luciérnaga será atraída por otras luciérnagas independientemente de su sexo.
2. La atracción es proporcional a su brillo (intensidad luminosa), así para cualesquiera dos luciérnagas, la de menor brillo se moverá hacia la de mayor brillo. El atractivo es proporcional al brillo y ambos decrecen con el incremento de la distancia. Si no hay una que brille más respecto a otras, estas no se moverán aleatoriamente.
3. El brillo de una luciérnaga es afectado por la definición de la función objetivo.

Considerando lo anterior tenemos el pseudo-código del algoritmo luciérnaga:

Función Objetivo $f(x)$, $x = (x_1, \dots, x_d)^T$
 Generar población inicial de luciérnagas x_i ($i = 1, 2, \dots, n$)
 Intensidad luminosa I_i para x_i es determinada por $f(x_i)$
 Definir coeficiente de absorción de luz γ
 Mientras ($t < \text{Máx. generación}$)
 Para $i = 1: n$ para " n " luciérnagas
 Para $j = 1: n$ para " n " luciérnagas
 Si ($I_i < I_j$)
 mueve luciérnaga " i " hacia " j ";
 Fin Si
 Varía el atractivo con la distancia r
 vía $\exp(-\gamma * r)$
 Evaluar nuevas soluciones y actualizar
 intensidad luminosa
 Fin Para j
 Fin Para i
 Ordenar las luciérnagas y encontrar el mejor actual
 global g^*
 Fin Mientras
Post procesamiento de resultados y visualización

Fig. 4. Pseudo-código de Algoritmo Luciérnaga

Se tienen dos consideraciones importantes: la variación de la intensidad luminosa y la formulación del atractivo.

Simplificadamente la intensidad luminosa $I(r)$ varía según:

$$I(r) = \frac{I_s}{r^2} \quad (20)$$

Donde I_s es la intensidad de la fuente, para un medio con un coeficiente de absorción luminosa fijo: γ , la intensidad luminosa varía según:

$$I = I_0 * e^{-\gamma * r} \quad (21)$$

A fin de evitar la singularidad en $r=0$ y para cumplir (20) y (21) se tiene:

$$I(r) = I_0 * e^{-\gamma * r^2} \quad (22)$$

Como el atractivo de la luciérnaga es proporcional a la intensidad luminosa, entonces se define:

$$\beta(r) = \beta_0 * e^{-\gamma * r^2} \quad (23)$$

Donde β_0 es el atractivo a una distancia igual a cero.

Se define a “r” como la distancia cartesiana o Euclídea entre dos luciérnagas.

Así se define el movimiento de una luciérnaga “i” que es atraída a otra luciérnaga “j” más atractiva (brillante):

$$x_i = x_i + \beta_0 * e^{-\gamma * r_{ij}^2} * (x_j - x_i) + \alpha * \epsilon_i \quad (24)$$

Donde el segundo término es atribuido a la atracción, el tercero es a la aleatoriedad con α como parámetro de aleatorización y ϵ_i un vector de números aleatorios.

B. Algoritmo Recocido Simulado

En 1983 apareció el algoritmo de recocido simulado, una técnica de optimización propuesta por S. Kirkpatrick, C. D. Gellat and M. P. Vecchi, inspirada en el proceso de recocido que se utiliza en la industria metalúrgica para obtener metales con una estructura cristalina más uniforme y que permita eliminar los defectos en ella, mejorando determinadas propiedades. Dicho proceso consiste en fundir el metal, donde la energía del sistema es máxima. Luego se hace descender la temperatura muy lentamente por etapas, dejando que en cada una de esas etapas los átomos queden en equilibrio (energía del sistema es mínima).

El RS es un algoritmo de búsqueda aleatoria basado en la trayectoria y una de sus principales ventajas es la habilidad para evitar caer en puntos críticos locales, de hecho se ha demostrado que este algoritmo podría converger a su óptimo global si se da una adecuada aleatoriedad y lento tiempo de enfriamiento.

Durante la ejecución del algoritmo, se nota que en cada iteración no siempre elegirá a un vecino que mejore la solución actual, sino que podría seleccionar una no correcta, la aceptación o no dependerá de una función probabilística que depende de dos factores: (1) la diferencia entre el estado actual y el nuevo; y (2) la temperatura. La función de probabilidad que se considera es [14], [15]:

$$p = e^{\frac{\text{eval}(E_{\text{actual}}) - \text{eval}(E_{\text{nuevo}})}{T}} \quad (25)$$

Dónde:

Eactual: estado actual

Enuevo: estado nuevo

Eval: función de evaluación o función objetivo (cambio de nivel de energía)

T: temperatura

La Ec. (25) se basa en la distribución de Boltzmann en mecánica estadística. Esto da cuenta que para "T" altos habrá una mayor probabilidad de aceptar estados nuevos que no mejoren la solución y conforme disminuye "T" (enfría) este se estabiliza y disminuye notablemente la probabilidad de aceptar un estado nuevo erróneo.

Se deberá inicializar la temperatura inicial, temperatura final a la cual se detiene las iteraciones y la velocidad de enfriamiento según el diseño de nuestro problema.

La temperatura puede ir enfriándose linealmente pero se tiene otras opciones:

Descenso Exponencial: $T_{i+1} = \delta * T_i$ (26)

Criterio de Boltzmann: $T_i = \frac{T_0}{1+\log(i)}$ (27)

Criterio de Cauchy: $T_i = \frac{T_0}{1+i}$ (28)

En base a lo anterior tenemos el siguiente pseudo-código:

Función Objetivo $f(x)$, $x = (x_1, \dots, x_p)^T$

Definir: temperatura inicial T_0 y evaluación inicial $x^{(0)}$

Definir temperatura final T_f y máximo número de iteraciones N)

Definir tasa de enfriamiento $T \rightarrow \alpha * T$, ($0 < \alpha < 1$)

Mientras ($T > T_f$ y $n < N$)

Mueve aleatoriamente a nuevas posiciones:

$x_{n+1} = x_n + \epsilon$ (recorrido aleatorio)

Calcular $\Delta f = f_{n+1}(x_{n+1}) - f_n(x_n)$

Se acepta la nueva solución si es mejor

Si no mejora

Genera un número aleatorio "r"

Acepta si $p = \exp(-\Delta f/T) > r$

Fin Si

Actualiza el mejor x^* y f^*

$n = n + 1$

Fin Mientras

Fig. 5. Pseudo-código de Algoritmo Recocido Simulado

4. RESULTADOS DE LA UTILIZACIÓN DE ALGORITMOS METAHEURÍSTICOS EN LA SINTONIZACIÓN DE UN CONTROLADOR PID

Para la aplicación de los algoritmos previamente revisados se ha considerado la función objetivo o de coste (intensidad luminosa y función de evaluación de los algoritmos luciérnaga y recocido simulado respectivamente) al Índice de Comportamiento ITSE (integral del tiempo multiplicada por el error al cuadrado), dado que proporciona la mejor selectividad de los demás índices, el valor mínimo se discierne fácilmente a medida que varían los parámetros del sistema [4],[6], además de poder usarse para elevados errores en los primeros instantes dado que el problema es de regulación donde las perturbaciones generalmente serán de tipo impulso.

$$ITSE = \int_0^t t * e^2(t) * dt \quad (29)$$

Donde: e es el error entre el set point y la variable de proceso y t es el tiempo de análisis.

Se ha considerado el criterio de Routh-Hurwitz para definir el valor máximo del rango de los parámetros del controlador PID según la ecuación (19). El valor mínimo se ha definido en varios rangos para evaluar los algoritmos.

Para el Algoritmo Luciérnaga se tiene los siguientes parámetros:

TABLA I. Parámetros de Algoritmo Luciérnaga

Parámetro	Valor
Número de Luciérnagas:	60
Nº generación máxima:	100
Aleatoriedad: α	0.5
Intensidad inicial: β	0.2
Coefficiente de Absorción: γ	0.5

Para el Algoritmo Recocido Simulado se tiene los siguientes parámetros:

TABLA II. Parámetros de Algoritmo Recocido Simulado

Parámetro	Valor
Temperatura Inicial:	40
Temperatura Mínima:	0.05

Velocidad de Enfriamiento	20
Paso	0.05

En las figuras 6 a la 11 se presentan los resultados del Algoritmo Luciérnaga y como evolucionaron los óptimos de los parámetros del controlador conforme avanza las generaciones de luciérnagas.

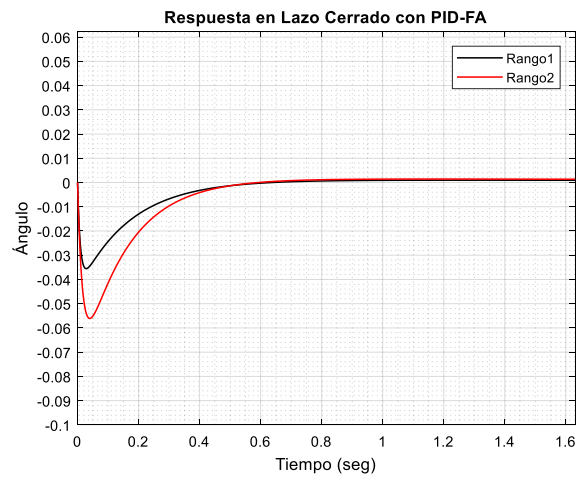
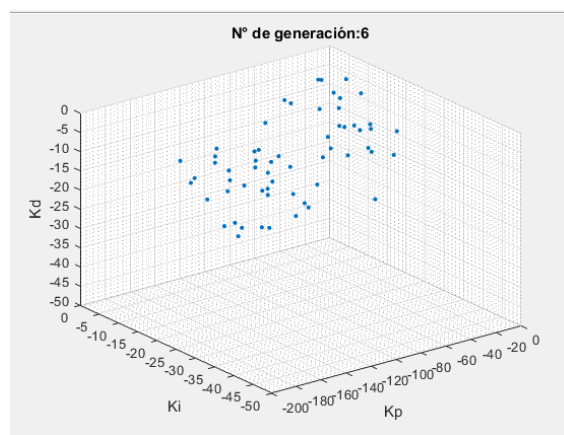
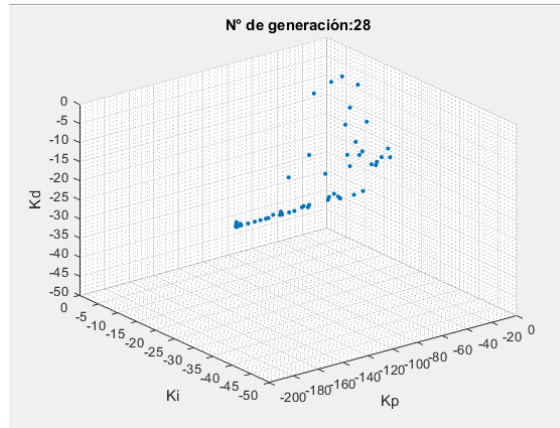


Fig. 6. Respuesta al impulso del lazo cerrado con FA



(a)



(b)

Fig. 7. Trayectoria de las luciérnagas para la obtención de los parámetros del controlador. (a) Inicio de la distribución de luciérnagas. (b) Convergencia de las luciérnagas

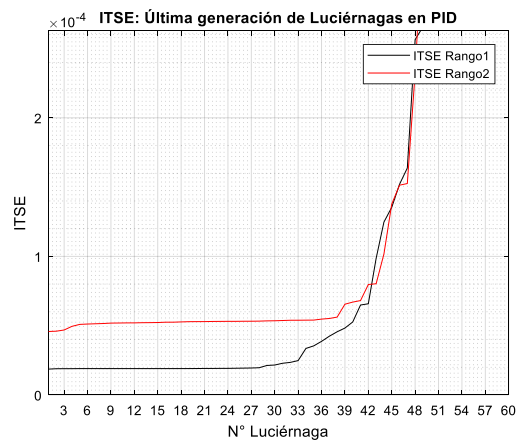


Fig. 8. Evolución del Índice de Comportamiento ITSE con Algoritmo Luciérnaga

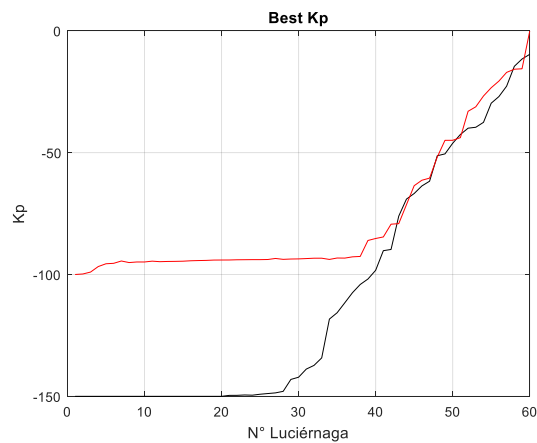


Fig. 9. Evolución del Kp (FA)

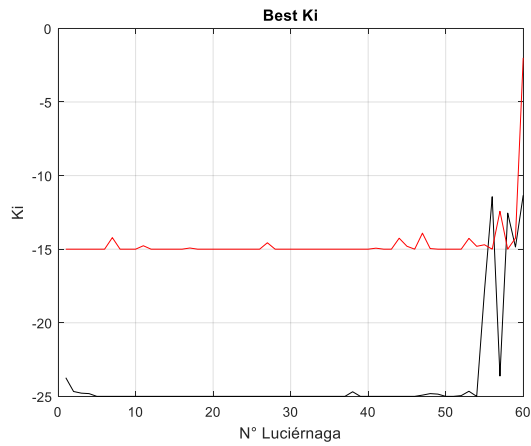


Fig. 10. Evolución de Ki (FA)

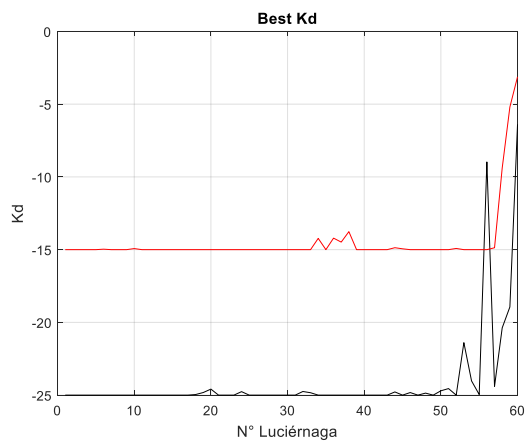


Fig. 11. Evolución de Kd (FA)

De las figuras 12 a la 16 se presentan los resultados para el Algoritmo Recocido Simulado. Para ambos algoritmos se verifica que los parámetros del controlador PID convergen.

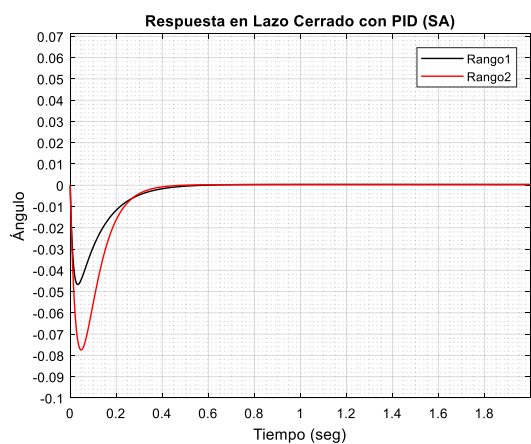


Fig. 12. Respuesta al impulso del lazo cerrado con SA

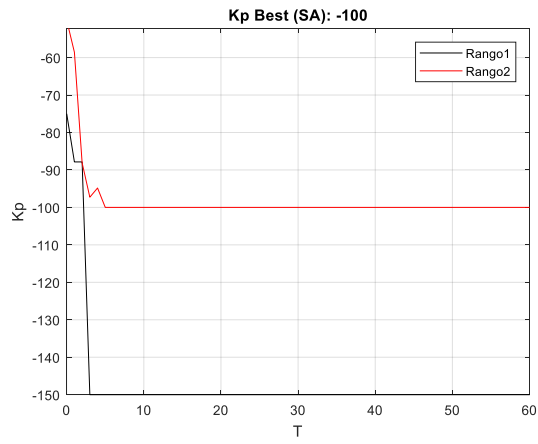


Fig. 13. Evolución de Kp (SA)

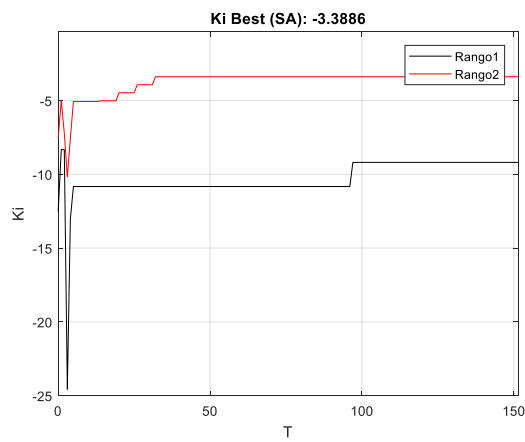


Fig. 14. Evolución de Ki (SA)

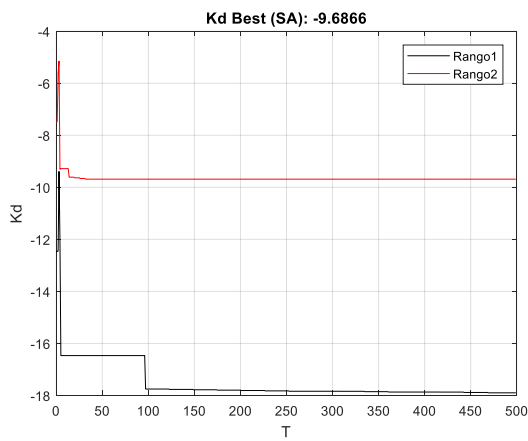


Fig. 15. Evolución de Kd (SA)

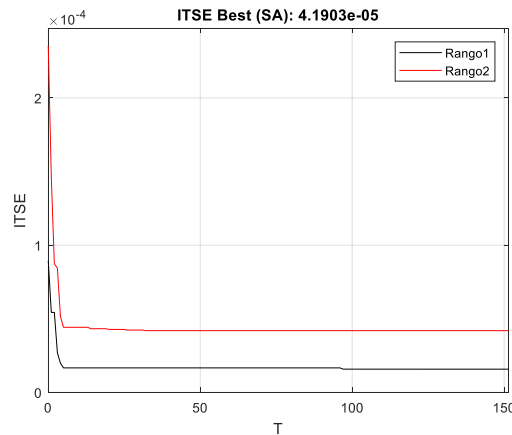


Fig. 16. Evolución del Índice de Comportamiento ITSE con Algoritmo Recocido Simulado

CONCLUSIONES

- Se observa que el índice de comportamiento ITSE está en el orden de $10e-4$ para ambos algoritmos y rangos, incluso para el Recocido Simulado se puede apreciar que los parámetros del controlador PID convergen más rápido y desde la primera iteración se tiene un significativo aporte, esto podría reflejarse en un menor tiempo de procesamiento de memoria en CPU.
- Si bien el Recocido Simulado alcanzó la estabilidad más pronto se tuvo un mayor tiempo de ejecución, por ello es importante probar si la velocidad de enfriamiento fue la más adecuada.
- Se verifica que es factible el uso de los algoritmos metaheurísticos para aplicación mediante un híbrido para la sintonía de lazos de control.
- Queda para futuros trabajos verificar el comportamiento de la señal de control (regulación) y evaluar si es factible alcanzar los resultados obtenidos (que tanto se podría sobre esforzar al actuador), en pruebas que no se muestran se identificó que se podría disminuir el índice de comportamiento a valores menores en un mayor rango de búsqueda y tiempo de ejecución.
- Se podría intentar una función objetivo (intensidad luminosa y de evaluación para el A. Luciérnaga y Recocido Simulado respectivamente) que considere los parámetros que estarían involucrados en el párrafo anterior para brindar un set específico de especificaciones de diseño.

AGRADECIMIENTOS

A la Universidad Nacional de Ingeniería, sección de Posgrado de la Facultad de Ingeniería Eléctrica y Electrónica por los conocimientos impartidos y fomentar el espíritu de la investigación.

REFERENCIAS

- [1] C. W. Anderson, "Learning to control an inverted pendulum using neural networks", *IEEE Control Systems Magazine*, vol. 9, no. 3, pp. 31-37, Apr. 1989.

- [2] N. Surendranath, S. Saketh y P. Pal, “Optimal PID controller design of an inverted pendulum dynamics: a hybrid pole-placement & firefly algorithm approach”, *IEEE First International Conference on Control, Measurement and Instrumentation*, vol. 1, no. 1, pp. 305-310, Jan. 2016.
- [3] M. Magaña, F. Holzapfel y “Fuzzy-logic control of an inverted pendulum with visión feedback”, *IEEE Transactions on Education*, vol. 41, no. 2, pp. 165-170, May. 1998.
- [4] R. C. Dorf y R. H. Bishop, *Sistemas de Control Moderno*, España: Pearson Prentice Hall, 2005.
- [5] K. Ogata, *Ingeniería de Control Moderna*, España: Pearson Educación – Prentice Hall, 2010.
- [6] E. Umez-Eronini, *Dinámica de Sistemas de Control*, México: Thomson Learning, 2001.
- [7] MATLAB, (2016, Set 10). “Inverted Pendulum: System Modeling”, [Online]. Disponible en: <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling>
- [8] A. Rojas-Moreno, *Control Avanzado: Diseño y Aplicaciones en Tiempo Real*, Perú: Editorial Universitaria de la Universidad Nacional de Ingeniería, 2001.
- [9] J. Brownlee, (2016, Set 10). “Clever Algorithms: Nature-Inspired Programming Recipes”, [En línea]. Disponible en: <http://www.cleveralgorithms.com/> Ver también el link: <https://archive.org/details/CleverAlgorithms>
- [10] C. Blum y A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”. *ACM Computing Surveys*, vol. 35, no. 3, pp. 268-308, Set. 2003.
- [11] T. Weise, (2016, Set 10) “Global Optimization Algorithms: Theory and Application”, [En línea]. Disponible en: <http://www.it-weise.de/projects/book.pdf>
- [12] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, United Kingdom: Luniver Press, 2010. [En línea]. Disponible en: https://www.researchgate.net/publication/235979455_Nature-Inspired_Metaheuristic_Algorithms
- [13] X. S. Yang, “Firefly Algorithm, Lévy Flights and Global Optimization”, *Research and Development in Intelligent Systems XXVI*, pp. 209-218, 2009.
- [14] A. García, *Inteligencia Artificial: Fundamentos, Prácticas y Aplicaciones*, México: Alfaomega Grupo Editor, 2016.
- [15] K.-L. Du y M. N. Swamy, *Search y Optimization by Metaheuristic: Techniques and Algorithms Inspired by Nature*, Switzerland: Birkhauser-Springer International Publishing, 2016.

Recibido (Received): 15 / 03 / 2019

Aceptado (Accepted): 22 / 06 / 2020