Diseño de controlador proporcional-integral-derivativo de celda termoeléctrica mediante algoritmo genético con codificación real

Design of a proportional-integral-derivative controller of thermoelectric cell with a genetic algorithm by real coding

Juan Fernando García-Mejía [1, 2*] Agustín Pérez-Guadarrama,[1] Laura Colín-Rivas[3]

[1] Centro Universitario UAEM Atlacomulco.
Carretera Toluca-Atlacomulco km 60. Atlacomulco, Estado de México, México
[2] Universidad Popular Autónoma del Estado de Puebla.
21 sur 1103, Barrio Santiago. Puebla, México. CP 72410
[3] Instituto Tecnológico de Toluca.
Av. Tecnológico Ex Rancho La Virgen. Metepec, Estado de México, México
* Correo-e: fgarciam@uaemex.mx

PALABRAS CLAVE:

algoritmo genético, celda termoeléctrica, codificación real

RESUMEN

Una celda termoeléctrica es un dispositivo semiconductor usado en la refrigeración, y un intercambiador de temperatura eléctrico. Suele mejorarse su desempeño por medio de un dispositivo denominado controlador proporcional, integral derivativo, cuyo cálculo se realiza por medio del método conocido como criterio de Nichols. En este artículo se propone una alternativa de cálculo, basada en un algoritmo genético con codificación real, el cual mejora el error cuadrático medio y el tiempo de establecimiento del dispositivo termoeléctrico. La propuesta presentada se simula en Scilab.

KEYWORDS:

genetic algorithm, thermoelectric cell, real encoding

ABSTRACT

A thermoelectric cell is a semiconductor device used in refrigeration, since it is an electrical temperature exchanger, enhancing its performance is usually done by a device called proportional-integral-derivative driver, which is calculated by a method known Nichols criterion. This paper proposes an alternative calculation, based on a real-coded genetic algorithm, which improves the mean square error and settling time of the thermoelectric device. The proposal presented is simulated in Scilab.

1 INTRODUCCIÓN

En esta sección se describen los conceptos que son utilizados en el presente trabajo: primero se define optimización, posteriormente se describe el algoritmo genético simple, y por último se presenta la información sobre el comportamiento matemático de una celda termoeléctrica.

1.1 Optimización

La optimización es un área de la ingeniería y un conjunto de ideas que encuentran el mejor resultado bajo ciertas consideraciones denominadas restricciones [1]. Un caso particular es la optimización numérica, proceso en el cual el óptimo global (valor máximo o mínimo) de una función objetivo es buscado mediante la elección de una combinación de variables. Esto se puede expresar en la ecuación (1) [2].

$$\min(\max)\, f(x), x = [x_1, x_2, \dots x_n]^T \in \mathbb{R}^n \quad (1)$$

Las restricciones son un problema que suelen expresarse como

$$g_j(x) \leq 0, \qquad j = 1,2,\dots,m$$
$$h_j(x) = 0, \qquad j = 1,2,\dots,r$$

donde $f(x), g_j(x)$ y $h_j(x)$ son funciones escalares del vector x. Los componentes de $x = [x_1, x_2, \dots x_n]^T$ son denominados variables, $f(x)$ es la función objetivo, $g_j(x)$ y $h_j(x)$ son funciones que describen las condiciones de iniquidad e igualdad respectivamente. El vector óptimo x que resuelve la expresión (1) se denota por x^* con el correspondiente valor de optimización $f(x^*)$. Las posibles formas de resolver el problema general descrito en (1) son las siguientes [3]:

i. analíticamente
ii. emulándolo y midiendo físicamente las variables de interés
iii. resolviéndolo mediante técnicas computacionales, como los algoritmos evolutivos

Los algoritmos evolutivos (EA, por sus siglas en inglés) son series de pasos definidos con aplicación en la optimización o de aprendizaje con la capacidad de evolucionar. Estos tienen tres características principales [4]:

i. Basado en la población. Un EA mantiene un conjunto de soluciones llamado población.
ii. *Fitness*-orientada. Toda la solución de una población se llama individuo, tiene una representación llamada código y la evaluación de desempeño se denomina aptitud (o *fitness*).
iii. Variación de motor. Los individuos de una población que se someten a una serie de operaciones con la finalidad de introducir variaciones en su código.

1.2 Celda termoeléctrica

Una celda termoeléctrica (TEC, por sus siglas en inglés) es un dispositivo semiconductor que tiene la capacidad de generar calor o frío en función de la magnitud y polaridad de una corriente eléctrica de alimentación. Las aplicaciones de los TEC se encuentran en microrrefrigeración, refrigeradores móviles, y presentan ventajas como su tamaño reducido, así como el largo tiempo de vida útil y amigable con el ambiente [5].

Una celda termoeléctrica de manera típica está caracterizada por la función de transferencia descrita en la ecuación (2) [6].

$$\tilde{T}_L(s) = G_1(s)\tilde{I}(s) + G_Q(s)\tilde{Q}_L(s) + G_a(s)\tilde{T}_a(s) \quad (2)$$

donde $\tilde{T}_L(s)$ representa la temperatura de la cara de enfriamiento. $\tilde{I}(s)$ es la corriente de entrada. $\tilde{T}_a(s)$ es la temperatura ambiente.

Además,

$$G_I(s) = \frac{N(s)}{sD(s)}$$

$$G_Q(s) = \frac{E_H \sinh(qL) + Akq\cosh(qL)}{D(s)}$$

$$G_a(s) = \frac{AA_F\, hkq}{D(s)}$$

donde

$$N(s) = \{Akq\,[\alpha_L \bar{T}_L \cosh(ql) - \alpha_H \bar{T}_H] + \alpha_L \bar{T}_L E_H \sinh(qL)\}s + \frac{Akq\beta}{C\gamma}[E_H(1 - \cosh(pL) - Akp\sinh(pL]$$

y $D(s) = AkqE_L\cosh(qL) + E_H E_L \sinh(qL) + AkqE_H\cosh(pL) + A^2k^2pq\sinh(pL)$

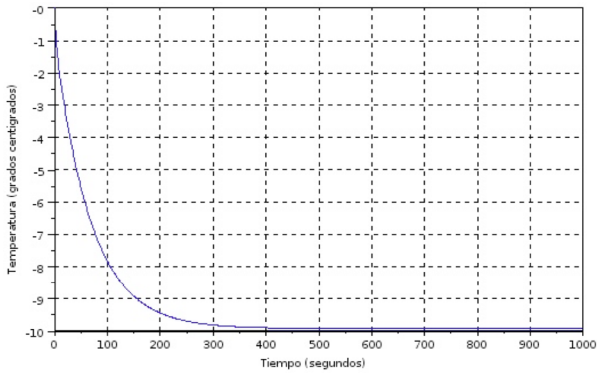

Figura 1. Respuesta de la celda termoeléctrica

El modelo matemático se reduce a la expresión mostrada en la ecuación (3), dado que se considera la temperatura ambiente como constante. Al aplicar las constantes especificadas por Mitchell [6] se obtiene la ecuación (4), cuya respuesta a una entrada de 1.55 amperios se muestra en la figura 1. Se puede observar que el tiempo de establecimiento, es decir, donde no se presentan cambios en el eje del tiempo, es de alrededor de 325 segundos, en un valor de -9.9º C en el eje y , que representa la temperatura de enfriamiento de la TEC.

$$\tilde{T}_L(s) = G_I(s)\tilde{I}(s) \qquad (3)$$

$$G(s) = -6.4061\left(\frac{0.064s + 0.00854}{s^2 + 0.5964s + 0.00855}\right) \qquad (4)$$

1.3 Desempeño de una TEC como un problema de ingeniería de control

El desempeño de la TEC mostrado en la figura 1 puede mejorarse mediante el uso de técnicas de ingeniería de control, disciplina que se encarga del estudio de esquemas que manipulen las respuestas de fenómenos físicos mediante actuadores. En este sentido, uno de los esquemas más usados es lo que se denomina controlador proporcional, integral, derivativo (PID), cuya expresión matemática se muestra en la ecuación 5, y la del TEC controlado por medio de un PID se muestra en la ecuación 6, donde $R(s)$ es la referencia de temperatura que se desea obtener.

$$C(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_p + k_i + k_d s^2}{s} \qquad (5)$$

$$\frac{\tilde{T}_L(s)}{R(s)} = \frac{G_I(s)\,C(s)}{1 + G_I(s)\,C(s)} \qquad (6)$$

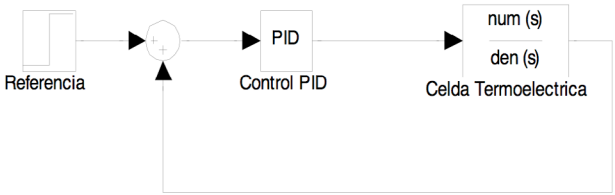

Figura 2. Esquema de control propuesto

Lo expresado en la ecuación 6 puede representarse con la figura 2; por otra parte, la forma más común de calcular los valores k_p, k_i, k_d es por medio del ajuste de Zigler-Nichols. Una alternativa de lo anterior es mediante un algoritmo genético, transformando un problema de control en uno de optimización, tal como se muestra en la sección 1.1.

2 ALGORITMO GENÉTICO

Uno de los primero algoritmos evolutivos empleados en la optimización numérica fue el algoritmo genético simple, desarrollado por John Holland en la década de los sesenta en la Universidad de Michigan [7]. Un algoritmo genético simula algunos aspectos propios de la teoría de la evolución de las especies de Darwin. Los mejores individuos de una determinada población tienen mayores posibilidades de supervivencia y reproducción; las posibilidades disminuyen o son nulas para los débiles.

Los pasos que caracterizan a un algoritmo genético simple son los siguientes [8, 9]:

i. definir una función de aptitud o función objetivo f_{obj}
ii. generar una serie de posibles soluciones de manera aleatoria (población)
iii. codificar la población
iv. evaluar f_{obj} con la población, iniciando así la i esima generación
v. seleccionar soluciones que se reproducirán
vi. aplicar una operación de cruzamiento
vii. mutar algunos elementos resultantes del cruzamiento
viii. reemplazar elementos de la población de la i esima generación con los mejores elementos de vi y vii
ix. detener, si se cumple criterio de paro, y en caso contrario saltar a iv

La población, como se indica en el paso ii del listado anterior, está formada por posibles soluciones del problema que reciben el nombre de cromosomas y que se codifican de diversas maneras. Las unidades mínimas existentes en éste se denominan gen. Algunas formas de codificación de cromosomas se enlistan a continuación:

i. codificación binaria
ii. codificación alfanumérica
iii. codificación real

Otro aspecto relevante de los algoritmos genéticos son los operadores involucrados, que son:

i. Selección. Se encarga de escoger pares de cromosomas que se cruzarán.
ii. Cruzamiento. Recombina las características de dos cromosomas para obtener una nueva solución. Este operador depende del tipo de codificación seleccionado.
iii. Mutación. Se encarga de variar de manera aleatoria algunos elementos obtenidos del cruzamiento (menos de 5% de la población), pero depende de la codificación, al igual que el cruzamiento.

3 METODOLOGÍA

En esta sección se muestran los pasos que se realizaron para optimizar el controlador PID, aplicado a la celda termoeléctrica. El objetivo es la minimización del error medio cuadrático, el cual se explica como la diferencia entre la respuesta del TEC y la referencia deseada.

3.1 Función objetivo

Como se mencionó en secciones anteriores, un algoritmo genético tiene una función objetivo, y ésta únicamente puede ser maximizada. Para construir la función objetivo se parte de la ecuación (6) y se obtiene la (7).

$$\hat{T}_L(s) = \frac{G_l(s)\,C(s)}{1 + G_l(s)\,C(s)} R(s) \quad (7)$$

De (7) y (5) se puede observar que la salida $\hat{T}_L(s)$ depende de los valores k_p, k_i, k_d, y a partir de la definición clásica del error se puede definir la ecuación (8):

$$e(s) = R(s) - \hat{T}_L(s) \quad (8)$$

De (8) puede obtenerse una función objetivo que se muestra en la ecuación (9)

$$f_{obj}(k_p, k_i, k_d) = \max\left(\frac{1}{1 + \sqrt{\frac{1}{T}\int_0^T e(k_p, k_i, k_d)^2}}\right) \quad (9)$$

3.2 Población y codificación

La colección de sujetos propuestos como posibles soluciones son generados de manera aleatoria (40 en total) con una distribución uniforme, codificando los cromosomas con números reales, de tal forma que el formato del cromosoma tiene la siguiente forma:

$$cromosoma = [k_p, k_i, k_d]^T$$

3.3 Selección

Los cromosomas que se seleccionaron para el cruzamiento en sucesivas generaciones fueron escogidos mediante una ruleta, el más estocástico de los métodos de selección en relación con el torneo y el elitismo. Es por eso que fue empleado en este trabajo.

3.4 Cruzamiento

El cruzamiento, como fue explicado en la sección anterior, es determinado por el tipo de codificación. En este caso se emplea el operador de cruce denominado $BLX\alpha$ cuyo procedimiento se muestra en esta sección.

Si se tienen dos cromosomas $C_1 = [k_p^1, k_i^1, k_d^1]$ y $C_2 = [k_p^2, k_i^2, k_d^2]$ que fueron seleccionados mediante un procedimiento de ruleta, los descendientes de éstos $H_k = [k_p^k, k_i^k, k_d^k]$, donde $k = 1,2$, son generados de manera aleatoria con una distribución uniforme dentro del intervalo indicado en la expresión (10), para $\alpha = [0,1]$ [10]:

$$random([C_{min} - I\alpha, C_{max} + I\alpha]) \quad (10)$$

donde $C_{max} = max\langle C_i^1, C_i^2\rangle$ $C_{max} = min\langle C_i^1, C_i^2\rangle$

$$I = C_{max} - C_{min}$$

El valor seleccionado es $\alpha = 0.2$

3.5 Mutación

En la mutación se alternan de manera aleatoria dos individuos por cada generación. Para esto se utiliza el operador genético de mutación con distribución normal, cuyo proceso se muestra a continuación.

A partir de un cromosoma C_i' se puede obtener un cromosoma transformado o mutado C_i'' desde la siguiente expresión, donde el tamaño de paso de la mutación $\beta = [0,1]$ y la dirección de la misma se representa por d [11]:

$$C_i'' = C_i' + \beta * d \qquad (11)$$

4 RESULTADOS

Con base en lo anterior se escribió un programa en Scilab, un *software* matemático de licencia GNU cuya ejecución en Ubuntu 12.04 permite realizar simulaciones de manera más rápida en comparación con alternativas cuya ejecución se realiza sobre sistemas operativos Windows; para este experimento se utilizó una computadora con una memoria de 3 GB en RAM. En esta sección se discuten los resultados del desarrollo que se documenta en esta propuesta. En la figura 3 se muestran las soluciones k_p, k_i, k_d generadas de manera aleatoria con un distribución normal. El rombo representa el óptimo encontrado, la figura 4 muestra en la línea azul continua la respuesta obtenida por medio del PID ajustado con algoritmos genéticos, en línea azul punteada se muestra el ajuste de Nichols y la línea verde representa la referencia. En la tabla 1 se muestra la síntesis de resultados.

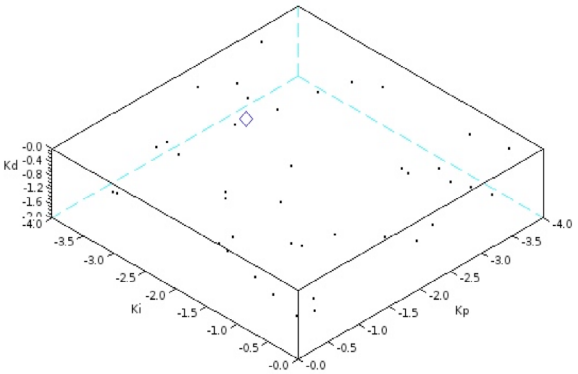

Figura 3. Soluciones generadas y solución encontrada

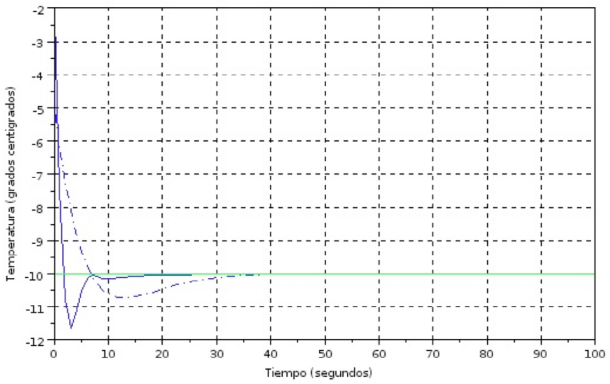

Figura 4. Respuesta de Nichols frente a respuesta ajustada con algoritmo genético frente a referencia

Tabla 1. Síntesis de resultados

Parámetro	Ajuste genético	Ajuste de Nichols
Tiempo de establecimiento	25 seg	36 seg
Ganancias k_p, k_i, k_d	*[2.5,3.2,1.6]*	*[2.4,0.6,2.4]*
Sobre impulso	11.6	10.7
Error cuadrático medio	0.605	0.675

5 CONCLUSIONES

Como lo muestran los resultados mostrados en la tabla 1, el tiempo de establecimiento, así como el error cuadrático medio, se reducen, que era el objetivo del algoritmo genético. El sobreimpulso presenta un incremento, que puede ser disminuido mediante el uso de restricciones o por medio de múltiples objetivos. Es necesario usar otros algoritmos evolutivos, con la finalidad de realizar un análisis comparativo. Como trabajo a futuro es recomendable el uso de otras heurísticas, como el enjambre de partículas o los algoritmos inmunológicos.

Actualmente se está ensamblando el circuito de potencia para implementar la aplicación de un refrigerador en miniatura con esta tecnología.

AGRADECIMIENTOS

Los autores agradecen a la Secretaria de Investigación y Estudios Avanzados de la Universidad Autónoma del Estado de México por el apoyo brindado para la realización de este proyecto.

REFERENCIAS

1. Chow Lee K. (2008). The success of information security projects: An investigation of the project management process, project cost, project risks and user acceptance.
2. Ponstein J. P. (2004). *Approaches to the theory of optimization*. Cambridge University Press.
3. Snyman, J. A. (2005). Practical mathematical optimization: *An introduction to basic optimization theory and classical and new gradient-based algorithms*. Birkhäuser.
4. Burger, C. (2007). *Propeller performance analysis and multidisciplinary optimization using a genetic algorithm*. Auburn University.
5. Tarter, R. (1993). *Solid-state power conversion handbook*. John Wiley and Sons.
6. Song Shaojing. (2010). Temperature control of thermoelectric cooler based on adaptive NN-PID International Conference on Electrical and Control Engineering.
7. Yang, X. S. (2011). *Nature-inspired metaheuristic algorithms*. Luniver Press.
8. Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT Press.
9. Mitsuo Gen, Runwei Cheng. (2000). *Genetic algorithms and engineering optimization*. John Wiley and Sons.
10. Sánchez, A. M. Algoritmos genéticos para codificación real con operador de cruce híbrido con múltiples descendientes: 2BLX0.5-2FR0.5-2PNX3-2SBX0.01 VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados.
11. King, I. (2006). *Neural information processing*. Springer.

Acerca de los autores

Juan Fernando García Mejía es maestro en Ciencias de la Electrónica por el Instituto Tecnologico de Toluca. Es profesor de tiempo completo del Centro Universitario UAEM Atlacomulco. Su línea de investigación es la intrumentación virtual y la optimización numérica.

Agustín Pérez Guadarrama es licenciado en Informática Administrativa. Realizó sus estudios en el Centro Universitario UAEM Atlacomulco. Funge como servidor público en la Secretaría de Desarrollo Social. Sus áreas de interés son inteligencia artificial, administración, arquitectura computacional.

Laura Colín Rivas es ingeniera en Computación por el Centro Universitario UAEM Atlacomulco. Actualmente es la encargada de salas de cómputo de la misma institución. Su línea de interés es el control inteligente.