**Sintonización de parámetros PID basada en algoritmos genéticos usando un modelo de Elastic Net: un estudio de caso de control de velocidad de un motor de CC**

**Genetics Algorithms based PID tuning using a Elastic Net model: A dc motor velocity control case study**

**RESUMEN**

El presente trabajo trató la aplicación de la metodología de identificación de sistemas para obtener un modelo de Elastic Net y una función de transferencia de un motor de CC; y la inclusión de ambos modelos en sendos procesos de sintonización de un controlador PID de velocidad basados en algoritmos genéticos, tomando como función objetivo a la integral del tiempo multiplicado por el valor absoluto del error. Así también, se utilizó la técnica de Ziegler-Nichols (primer y segundo método) y la aplicación PID Tuner de Matlab para sintonizar tres controladores PID adicionales. Posteriormente, se comparó el desempeño de los controladores obtenidos, observándose un mejor rendimiento en el caso del uso del modelo de Elastic Net. Además, este modelo demostró abstraer el comportamiento del motor de CC mejor que la función de transferencia, ya que por su misma naturaleza, esta última no puede captar el comportamiento no lineal del motor de CC, sobre todo a bajas velocidades.

**Palabras clave:** algoritmos genéticos, control PID, Elastic Net, función de transferencia, identificación de sistemas

**ABSTRACT**

**1. INTRODUCCIÓN**

Más del 90% de todos los circuitos de control son PID (Åström y Hägglund 2001). Este tipo de control ha sido el método predominante desde el inicio de los sistemas de automatización (Vilanova y Visioli 2012 p. 460) y es el algoritmo de control más comúnmente utilizado en la industria actual(Kocur, Kozak y Dvorscak 2014).

La investigación en esta área se concentra principalmente en el desarrollo de metodologías de ajuste de los tres parámetros PID (Vilanova y Visioli 2012 p. 320), y aún la mayoría de documentos sobre control tratan el método de Ziegler-Nichols como punto de referencia para sintonizar controladores PID. Esta es una situación muy insatisfactoria porque las reglas de Ziegler-Nichols son conocidas por dar resultados muy pobres en muchos casos (Vilanova y Visioli 2012 p. 416). Por otro lado, Marin Cano, Hernandez Rivero y Jimenez Builes (2018); y Colorado Arellano et al. (2018) muestran controladores sintonizados por medio de algoritmos evolutivos que superan de forma amplia los métodos clásicos. Así también, por ejemplo, la aplicación de algoritmos bioinspirados en problemas de control como la sintonización basada en algoritmos genéticos permitió hallar un controlador PID óptimo (Meena y Devanshu 2017). Entonces es notorio el avance en técnicas de ajuste de parámetros PID, y las técnicas mencionadas comparten la característica de que utilizan una función de transferencia como resultado de la identificación de un determinado sistema dinámico. Sin embargo, los enfoques tradicionales basados en modelos, que requieren conocer previamente el proceso o modelo físico obtenido a partir de principios primarios, se han vuelto poco prácticos, especialmente para la industria a gran escala (Yin et al. 2015). Frente a esto, cabe mencionar que la aplicación de métodos como máquinas de soporte vectorial, LASSO o redes neuronales ha mostrado resultados sobresalientes, respecto de métodos tradicionales, sobre modelos de identificación de sistemas tipo NARMAX o sistemas lineales.

Y encontrar modelos capaces de abstraer incluso comportamientos no lineales de sistemas dinámicos es una oportunidad para sacar mejor provecho de técnicas de optimización aplicadas al control. Además, a mejores controladores siempre corresponde un incremento en la eficiencia y eficacia en los procesos industriales.

Por lo tanto, se presenta un escenario idóneo para integrar modelos obtenidos por métodos modernos de identificación de sistemas en procesos de sintonización bioinspirados como los basados en algoritmos genéticos. Es así que con el objetivo de lograr dicha integración, se selecciona a Elastic Net como el modelo adecuado para identificar de forma no paramétrica a un motor de CC y posteriormente ser incluido en la sintonización basada en algoritmos genéticos de un controlador PID de velocidad.

**2. Materiales y métodos**

**2.1 Materiales**

**-** Motor <código> con encoder <código> acoplado mecánicamente

**Descripción:**

- Driver IRF520

**Descripción:**

- Laptop <código>

**Descripción:**

- Fuente de alimentación

**Descripción:**

- Arduino UNO R3

**Descripción:**

- Jupyter Notebook

**Descripción:**

- Arduino IDE

**Descripción:**

- MatLab:

**Descripción**

**2.2 Métodos**

El desarrollo experimental fue abordado en dos etapas:

**2.2.1 Etapa I: Adquisición y procesamiento de datos**

Esta etapa inicia con la selección de un motor de CC (sistema dinámico de interés) como planta de control y la implementación de un sistema de control de velocidad, sin perturbaciones externas y sometido solo a cambios repentinos del valor de consigna o setpoint. => Muestreo de datos => Accondicionamiento de datos =>

**a)** Sintonización basada en la técnica de Ziegler-Nichols (primer método) => Obtención de un controlador => [pid-1-Z-N]

**b)** Selección de una estructura de modelo por medio de =>

**b.1)** Elastic Net & **b.2)** La función de transferencia => Ajustar modelo => Validar modelo => Simulación del sistema de control usando el modelo => Elección de la estrategia de sintonización =>

**b.1)** [AG-EN] & **b.2)** [AG-FT, PID Tuner, Z-N (segundo método)] => Obtención de controladores => **b.1)** [pidAG-EN] & **b.2)** [pidAG-FT, pidTUNER, pid-2-Z-N]

**2.2.1.1 Muestreo de datos**

**2.2.1.2 Acondicionamiento de datos**

**2.2.1.3 Ajuste y Validación de modelos**

**2.2.1.3.1 Caso de la función de transferencia: usando la aplicación de identificación de sistemas de MatLab**

**2.2.1.3.2 Caso del modelo de Elastic Net: usando la librería SciKit-Learn de python en Jupyter Notebook**

**2.2.1.4 Simulación del sistema de control**

**2.2.1.4.1 Usando la función de transferencia**

**2.2.1.4.2 Usando el modelo de Elastic Net**

**2.2.2.1 Evaluación de desempeño de controladores PID en un contexto simulado**

**2.2.1.5 Sintonización de controladores PID**

**2.2.1.5.1 Técnica de Ziegler-Nichols (primer méodo)**

**2.2.1.5.2 Técnica de Ziegler-Nichols (segundo méodo)**

**2.2.1.5.3 Algoritmos genéticos**

**2.2.1.5.4 PID Tuner de MatLab**

**2.2.2 Etapa II: Control**

Implementación de cada grupo de parámetros PID sobre el sistema de control =>

**c)** [pid-1-Z-N]

**d)** [pidAG-EN]

**e)** [pidAG-FT]

**f)** [pidTUNER]

**g)** [pid-2-Z-N]

**=>** Evaluación de desempeño de los controladores => contrastación de resultados => discusión => conclusiones

**2.2.2.1 Evaluación de desempeño de controladores PID en un contexto real**

**3. Resultados y discusión**

Muestreo y acondicionamiento de datos + ajuste y validación del modelo de Elastic Net y la función de transferencia + las simulaciones del sistema de control + Discusión y presentación de los controladores PID resultados de los procesos de sintonización:

**c)** [pid-1-Z-N]

**d)** [pidAG-EN]

**e)** [pidAG-FT]

**f)** [pidTUNER]

**g)** [pid-2-Z-N]

**4. Conclusiones**

**5. Referencias bibliográficas**

**6. Anexos (opcional)**