



POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

# Práctica de algoritmos genéticos.

Juan Arquero Gallego, Luis Santana Escudero, David García Castaño

# Table of Contents

Descripción del sistema a controlar. ....	1
Descripción del algoritmo de control a utilizar. ....	2
Descripción del procedimiento de ajuste del controlador por medio de un AG. ....	3
Forma de generar la población inicial. ....	3
Operadores genéticos utilizados y detalles de implementación utilizados.. ....	3
Función evaluadora utilizada. ....	4
Criterio de finalización del algoritmo genético. ....	4
Resultados obtenidos. ....	5
Valoración personal de los algoritmos genéticos.. ....	6

# Descripción del sistema a controlar.

El sistema a controlar es un motor DC, el cual se utiliza en una amplia variedad de aplicaciones en las que se requiere controlar la velocidad de giro. Este tipo de motor es uno de los más utilizados en la industria debido a su sencillez y fiabilidad.

En términos técnicos, un motor DC es un tipo de motor eléctrico que funciona con corriente continua. Esto significa que la corriente eléctrica que alimenta al motor fluye en una sola dirección, lo que permite un control preciso de la velocidad del motor.

En el caso específico del sistema que se está controlando, se espera que el motor alcance una velocidad de referencia de 80 RPM (revoluciones por minuto). Para lograr esto, se utilizará un controlador PID (proporcional-integral-derivativo).

# Descripción del algoritmo de control a utilizar.

El controlador PID se compone de tres componentes: el componente proporcional, el componente integral y el componente derivativo. Cada uno de estos componentes tiene un efecto diferente en el sistema de control y se ajustan mediante tres parámetros,  $K_p$ ,  $K_i$  y  $K_d$  respectivamente.

El componente proporcional tiene un efecto directo en la salida del controlador, proporcional al error entre la referencia y la medición del sistema. El componente integral tiene un efecto acumulativo en la salida del controlador, corrigiendo los errores residuales a lo largo del tiempo. El componente derivativo tiene un efecto anticipativo en la salida del controlador, anticipando los cambios futuros en el error del sistema.

En la solución proporcionada, se inicializa el controlador PID con los valores  $K_p$ ,  $K_i$  y  $K_d$  proporcionados como parámetros de entrada. Luego, en cada ciclo de simulación, el controlador PID ejecuta sus tres componentes y ajusta la salida del controlador en consecuencia. Esta salida es luego utilizada para actualizar el estado del sistema simulado, en este caso la velocidad del motor DC.

# Descripción del procedimiento de ajuste del controlador por medio de un AG.

El procedimiento de ajuste del controlador por medio de un AG se refiere al proceso de optimizar los parámetros del controlador utilizando un algoritmo genético. El algoritmo genético es un enfoque de optimización basado en la evolución natural, en el que una población de soluciones candidatas es sometida a procesos de selección, cruce y mutación para generar una nueva población en cada generación. El objetivo es encontrar la solución óptima para el problema de optimización.

En este caso, los parámetros del controlador a optimizar son  $K_p$ ,  $K_i$  y  $K_d$ , que son los tres parámetros utilizados en el algoritmo de control PID. Estos parámetros afectan directamente a la respuesta del sistema controlado y su optimización es esencial para lograr un desempeño óptimo del controlador.

## Forma de generar la población inicial.

La población consta de 2000 individuos que se genera de forma aleatoria. Cada individuo de la población se representa mediante un cromosoma, el cual contiene los valores de los parámetros  $K_p$ ,  $K_i$  y  $K_d$ . Los valores de estos parámetros están limitados dentro de un rango específico para evitar valores extremos.

## Operadores genéticos utilizados y detalles de implementación utilizados.

En el código proporcionado, se utiliza un algoritmo genético para ajustar los parámetros del controlador ( $K_p$ ,  $K_i$  y  $K_d$ ). Los operadores genéticos utilizados son la selección, la cruce y la mutación.

La selección se realiza mediante el uso de una función de selección de ruleta, donde los individuos con mejores puntuaciones tienen una mayor probabilidad de ser seleccionados para la reproducción conocidos como la élite.

La cruce se realiza mediante el uso de un operador de cruce de un punto, donde dos individuos seleccionados al azar se cruzan en un punto aleatorio para generar dos nuevos individuos.

La mutación se realiza mediante el uso de un operador de mutación gaussiano, donde se aplica una distribución gaussiana alrededor de cada parámetro de un individuo seleccionado al azar con una probabilidad predefinida para generar un nuevo individuo.

La implementación de estos operadores genéticos se realiza mediante un ciclo iterativo, donde se genera una nueva población a partir de la población actual mediante la aplicación de los operadores genéticos mencionados anteriormente. El proceso se repite hasta alcanzar un criterio de finalización, como un número de generaciones o una puntuación mínima.

## Función evaluadora utilizada.

La función evaluadora utilizada en este caso es la función Score, la cual se encarga de evaluar el desempeño del controlador. Los parámetros medidos son calculados en el cuerpo del procedimiento `Carry_out_a_simulation`, después de realizar la simulación. En primer lugar, se busca el tiempo de subida ( $T_r$ ) mediante un bucle for que recorre todos los valores de velocidad en la simulación. Se considera que el tiempo de subida ha ocurrido cuando se encuentra el primer valor de velocidad que es mayor o igual a la velocidad de referencia y el valor anterior es menor a la velocidad de referencia. Una vez encontrado el tiempo de subida, se busca el tiempo de pico ( $T_p$ ) mediante un bucle for que recorre los valores de velocidad desde  $T_r$  hasta el final de la simulación. Se considera que el tiempo de pico ha ocurrido cuando se encuentra el primer valor de velocidad que es menor o igual al valor anterior.

A continuación, se busca el tiempo de estabilización ( $T_s$ ) mediante un bucle for que recorre los valores de velocidad en orden inverso desde  $T_p$  hasta el final de la simulación. Se considera que el tiempo de estabilización ha ocurrido cuando se encuentra el primer valor de velocidad que cumple con la condición de estar dentro del 2% de la velocidad de referenciada.

Por último, el valor máximo de sobrepico ( $M_p$ ) se busca también mediante un bucle for que recorre todos los valores de velocidad en la simulación. Se considera que el sobrepico ha ocurrido cuando se encuentra el primer valor de velocidad que es mayor a la velocidad de referencia y menor al valor anterior.

Una vez que se han calculado estos cuatro parámetros ( $T_r$ ,  $T_p$ ,  $T_s$  y  $M_p$ ), se calcula la función Score como la suma de los valores absolutos de la diferencia entre los parámetros medidos y los parámetros esperados. Se espera que cuanto menor sea el valor de Score, mejor será el desempeño del controlador.

## Criterio de finalización del algoritmo genético.

La condición de finalización del algoritmo se define dentro del package `Simulator_operator` o `GA_operator`. Es el segundo paquete quien determina la finalización de éxito, el primero sirve como control para evitar una ejecución infinita.

Dentro del paquete `GA_operator`, el criterio de finalización del algoritmo genético está condicionado por dos expresiones booleanas excluyentes. La primera es el caso de éxito, que tiene lugar cuando el score obtenido por el primer individuo de la población es menor que 10. La segunda es que se hayan generado más de 500 generaciones sin haber conseguido la condición de éxito.

En el paquete `Simulator_operator`, la ejecución es abortada cuando alguno de los individuos genera una velocidad superior a 5 veces el valor de referencia o inferior a cero.

Dadas estas condiciones de salida se asegura que la ejecución del AG siempre finalice. Cabe destacar que el criterio de score inferior a 10 en la condición de éxito, la hemos manipulado para establecer diferentes límites modificando una variable al comienzo del código.

## **Resultados obtenidos.**

# **Valoración personal de los algoritmos genéticos.**