



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. ....	5
Resultados obtenidos. ....	6
Valoración personal de los algoritmos genéticos.. ....	8

# 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 angular de referencia de 80 rad/s (radianes por segundo). 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, por lo que afecta principalmente al comportamiento del sistema en régimen permanente. El componente derivativo tiene un efecto anticipativo en la salida del controlador, anticipando los cambios futuros en el error del sistema, por lo que afecta principalmente al error producido en régimen transitorio.

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 la simulación tomando estas tres componentes como valores del PID y ajusta la salida del controlador en consecuencia. La diferencia entre esta salida y la consigna establecida es el error, que es luego utilizado 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, y los parámetros  $K_p$  y  $K_i$  se generan de forma aleatoria para estos individuos, tomando valores entre 0 y 0.01 para  $K_p$ , y entre 0 y 0.00001 para  $K_i$ . Los valores de  $K_d$  se inicializan siempre a 0.00001. 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, no representativos de soluciones válidas. Se ha probado a realizar cambios en la generación de estos valores aleatorios para ajustarlos más a los valores que resultan tras la ejecución de la simulación. De esta manera, se obtienen resultados válidos en menos generaciones, pero el árbol de posibles soluciones se reduce, disminuyendo así la diversidad de los individuos de la población, lo que puede desembocar en un algoritmo menos adaptativo.

## 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 combinación y la mutación.

Se realizan tres combinaciones; entre individuos de la élite, entre un individuo de la élite y otro de la población general y entre individuos de la población general. Cada una de estas tres combinaciones conlleva un mecanismo de selección distinto. Para la combinación entre elementos de la élite, se escogen 2 individuos al azar pertenecientes al 10% de la población que ha obtenido mejores resultados. Para escoger elementos de población inicial, se escogen elementos al azar del total de población, incluyendo a la élite. Para realizar esta selección de forma cómoda, los distintos individuos se ordenan en un array, quedando los individuos de mejor puntuación primer (en los índices más bajos del mismo) y los individuos de peor puntuación al final. Así se harán accesos por secciones del array según el tipo de individuo que se quiera seleccionar.

La combinación entre dos individuos  $X$ ,  $Y$ , produce dos "hijos" para la próxima generación,

denotados  $Z1$ ,  $Z2$ . Estos cogen dos de los valores de su cromosoma (compuesto por  $Kp$ ,  $Ki$  y  $Kd$ ) del individuo  $X$  y el valor restante del individuo  $Y$ . Es equiprobable que este elemento sea  $Kp$ ,  $Ki$  o  $Kd$ . Este proceso es independiente del tipo de selección llevado a cabo.

La operación de mutación se realiza según el valor de tres variables aleatorias; parámetro, severidad, y signo. La primera indica si el parámetro a modificar será  $Kp$ ,  $Ki$  o  $Kd$ . La variable severidad indica como de significativa será la mutación, que implicará una suma o resta al parámetro seleccionado. A mayor severidad mayor es el valor con el que se realiza la operación, y por tanto, mayor es el cambio. La variable signo indica si se realizará una suma o una resta.

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, que será desarrollado a continuación.

## 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 ( $Tr$ ) 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. Es entonces cuando se considera que la función ha cruzado el valor de consigna. Esta comprobación respecto al valor anterior evita que se den como válidos  $Tr$  obtenidos de una función que sobrepasa en todo momento el valor consigna. Una vez encontrado el tiempo de subida, se busca el tiempo de pico ( $Tp$ ) mediante un bucle for que recorre los valores de velocidad desde  $Tr$  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, es decir, en el punto en el que la derivada de la función es igual o menor a 0 y por tanto se está produciendo una inflexión.

A continuación, se busca el tiempo de estabilización ( $Ts$ ) mediante un bucle for que recorre los valores de velocidad en orden inverso desde  $Tp$  hasta el final de la simulación. Se actualiza el tiempo de estabilización cuando se encuentra un valor de velocidad que cumple con la condición de estar dentro del 2% de la velocidad de referenciada. Este valor se deja de actualizar cuando la diferencia es mayor al 2%, de esta manera se obtiene el primer punto (desde el origen  $t=0$ ) en el que se ha llegado a dicho margen de validez.

Por último, el valor de máximo sobreimpulso ( $Mp$ ) se busca también mediante un bucle for que recorre todos los valores de velocidad en la simulación. Se considera que el sobreimpulso ha ocurrido utilizando el mismo criterio que para  $Tr$ . Además se realiza una penalización a los sistemas inestables que generan sobreimpulsos mayores al primero en las siguientes oscilaciones. Esto se hace mediante el guardado del primer  $Mp$  hallado, y la comparación del  $Mp$  correspondiente en el resto de puntos de inflexión con este primer  $Mp$ . En caso de que sean mayores se penaliza a este controlador aumentando el score.

También se realizan penalizaciones para  $Tr$ , en caso de que el sistema controlado no llegue en

ningún momento a la consigna establecida.

Una vez que se han calculado estos cuatro parámetros (Tr, Tp, Ts y Mp), se calcula la función Score como la suma de los valores absolutos de la diferencia entre los parámetros medidos y los parámetros esperados, que son calculados con anterioridad mediante métodos matemáticos que serán expuesto a continuación. 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 el valor definido para la constante "Criterio", que establece el máximo error permitido. En el caso de las ejecuciones presentadas, se ha establecido como criterio el valor de 5, que hace que el error sea igual o inferior a 1 en las distintas variables evaluadas (Tr, Tp, Ts y Mp). La segunda condición valorada es que se hayan generado más de 500 generaciones sin haber conseguido la condición de éxito, lo que supondría un mal ajuste del algoritmo.

En el paquete Simulator\_operator, la ejecución de cada una de las simulaciones es abortada cuando alguno de los individuos genera una velocidad superior a 5 veces el valor de referencia o inferior a cero. En este caso se considera el resultado de la simulación como no válido.

# Resultados obtenidos.

Tras la ejecución de nuestro programa en Ada, se ha obtenido el siguiente resultado:

```
14      0.014419041      0.007225362      0.000010062      10.169200029
14      2.218814291      0.006996235      0.000010006      4611686018.427387903
14      0.015228869      0.007245170      0.000010079      9.177139452
14      4.190396130      0.090466197      0.000000142      4611686018.427387903
14      0.015042543      0.027729443      0.000006685      392.747276810
14      0.017525249      0.006974527      0.000010153      15.673873852
14      2.705471520      0.089935051      0.000009993      4611686018.427387903
14      0.015164685      0.007245170      0.000009914      9.149164518
14      0.013647695      0.006286039      0.000006523      50.352530716
14      0.014797243      0.007011937      0.000028479      46.257138935
14      6.649247921      9.667530107      0.000009911      4611686018.427387903
14      0.016838830      0.007011937      0.000075540      135.595980263
14      0.015446293      0.007245170      0.000009992      10.278234018
14      0.019051272      0.006792458      0.000009987      34.578417549
14      0.019708474      0.007823501      0.000010000      46.090606411
14      0.012841814      0.007181851      0.000010029      14.873833428
14      0.017975174      0.007676327      0.000081751      91.069736923
14      0.014419041      2.935988980      0.000012019      4611686018.427387903
14      0.009813362      5.590569216      0.000054708      4611686018.427387903
14      0.000018146      0.006507709      0.000009877      52.182527371

The best controller is:

      0.014419041      0.006774304      0.000000746      4.801306103

Parámetros temporales obtenidos:
Tr:      137 Tp:      194 Ts:      291 Mp:      84.798693897

Generation:      14
[2023-01-15 22:21:56] process terminated successfully, elapsed time: 01:16.17s
```

Figure 1. Resultados del Algoritmo Genético.

Como se puede observar se ha modificado ligeramente la salida del mismo para poder visualizar los valores obtenidos de Tr, Tp, Ts y Mp tras la ejecución de la simulación en el programa. Estos valores están muy cerca de los esperados (Expected\_Tr= 140, Expected\_Tp=193, Expected\_Ts=291). Para averiguar estos valores "Expected", se ha utilizado la herramienta de cálculo simbólico Maxima. Un archivo .mxw será adjunto a esta memoria, en el que se detalla el procedimiento llevado a cabo para el cálculo, así como los resultados intermedios.

Los valores correspondientes a la configuración del PID son:

$K_p = 0.014419041$   $K_i = 0.006774304$   $K_d = 0.000000746$

Qué tras ser introducidos en scilab producen el siguiente resultado:

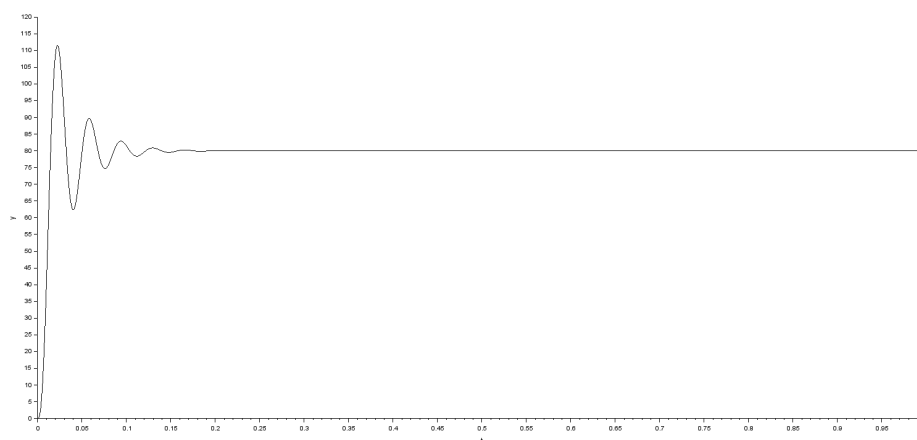


Figure 2. Simulación del resultado en Scilab.

Tras evaluar el resultado de la simulación se puede llegar a la conclusión de que el resultado obtenido por scilab no coincide con el resultado obtenido en la simulación llevada a cabo por el programa de Ada. Esto puede indicar que la simulación en scilab esté mal realizada, o en su



defecto que el cálculo realizado por la función de simulación en ADA no es representativo respecto al resultado de una implementación real del controlador. La simulación de scilab tiene el siguiente aspecto:

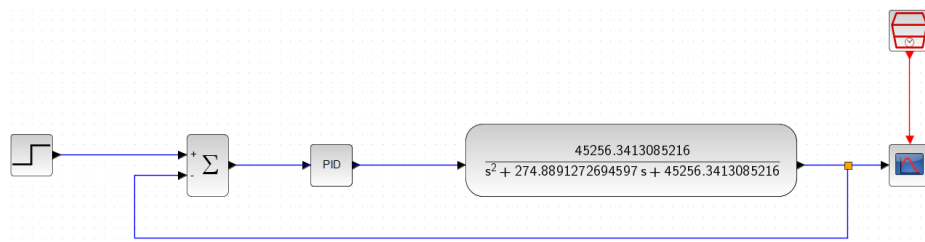


Figure 3. Diagrama de Scilab.

Como se puede observar, implementa una entrada escalón que va desde 0 a 80 (representando el punto de trabajo), seguidamente de un punto de suma. Tras este podemos encontrar el controlador PID, donde introduciremos los valores obtenidos del algoritmo genético, a excepción de  $K_i$  que introduciremos la inversa ( $1/k_i$ ). Seguidamente encontramos la función de transferencia, obtenida según los valores del enunciado y que se pueden observar los cálculos en el fichero de máxima.

# Valoración personal de los algoritmos genéticos.

Bajo nuestra experiencia, los algoritmos genéticos han resultado una forma muy eficiente de calcular los parámetros para controladores PID, pues exploran un gran número de posibles configuraciones de una forma automática, y se ajustan a distintos escenarios simplemente cambiando los valores de entrada "Esperados". Esto permite que dadas las entradas de  $T_r$ ,  $T_p$ ,  $M_p$ ,  $T_s$  y  $Controller\_Reference$ , el programa implementado pueda diseñar un PID mediante el ajuste iterativo de los valores de  $K_p$ ,  $K_i$  y  $K_d$ . En este caso, el modelo a ajustar es de segundo orden, y los resultados de ejecución se recogen en pocos segundos, lo cual resulta altamente eficiente en la vida profesional, permitiendo ahorrar mucho tiempo y recursos. Por otro lado, el entendimiento de este tipo de algoritmos ha sido bastante enriquecedor, ya que, para algunos de los miembros del grupo era la primera vez nos enfrentábamos con algo así, además de que la experiencia en programación en Ada era escasa. Aún así, el resultado es bastante satisfactorio, está claro que no es el ideal, pero como equipo la valoración final es positiva, considerandonos todos como satisfechos.