UNIVERSIDAD
CARLOS III DE MADRID

CONTROL INTELIGENTE

PRÁCTICA 1

Control de un motor de cc utilizando Algoritmos Genéticos

Introducción teórica al DE:

En esta práctica se va a realizar un control de posición para motor de corriente continua (de la función de transferencia que le describe el comportamiento) calculando los parámetros de un PID mediante algoritmos genéticos. Concretamente, se va a utilizar el Differential Evolution (DE), ideado por Kenneth Price y Rainer Storn en octubre de 1995.

El DE es un método de optimización perteneciente a la categoría de algoritmos genéticos que se usa para resolver problemas complejos. El algoritmo contiene un conjunto de candidatos a ser la solución del problema. Estos candidatos mutan y se recombinan para producir nuevos individuos. Estas nuevas soluciones serán admitidas en la solución según el valor de su función de coste.

Una de las características principales del algoritmo es el uso de vectores de prueba (recombinados y mutados) que compiten con los de la población en cada iteración. Sólo sobreviven los que tienen una mejor función de coste.

El algoritmo se basa en el vector de soluciones que contiene el número de variables a optimizar (D). La población está compuesta por NP vectores. Se define un vector x_i^k, donde *i* es el índice del individuo en la población (entre 1 y NP) y *k* es la generación o iteración correspondiente. Cada vector está a su vez compuesto de las variables del problema $x_{i,j}^k$(j entre1 y D), en donde *j* es el índice de la variable en el individuo

Las variables del problema están dentro de unos límites: x_j^{min} y x_j^{max}.

El algoritmo se compone básicamente de 4 pasos:.

- Inicialización
- Mutación
- Cruce
- Selección

La inicialización se realiza al principio de la búsqueda, y los pasos de mutación-cruce-selección se realizan para toda la población un número determinado de veces que vendrá dado por unas condiciones de convergencia o un límite superior de iteraciones. A continuación se detallan brevemente cada uno de estos pasos:

Inicialización

La población es inicializada (primera generación) aleatoriamente, considerando los valores mínimos y máximos de cada variable:

$$x_{i,j}^1 = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min})$$

para i=1…NP, j=1…D y siendo rand(0,1) un número aleatorio entre 0 y 1.

Mutación

La mutación consiste en la construcción de NP vectores mutados a partir de los elementos de la población actual. Hay distintas estrategias de mutación. Por ejemplo, una opción es partir de tres individuos aleatorios de la población:

$$v_i^k = x_a^k + F(x_b^k - x_c^k)$$

donde a, b y c son distinos entre sí. Se construyen NP elementos mutados en cada generación. F es un parámetro que controla la tasa de mutación, y se encuentra en el rango [0,2].

Cruce

Una vez obtenidos los NP vectores mutados, el cruce se efectúa de manera aleatoria, combinando los vectores originales con los mutados:

$$t_{i,j}^k = \begin{cases} v_{i,j}^k; & rand(0,1) < CR \\ x_{i,j}^k; & otro\ caso \end{cases}$$

para i = 1...NP, j = 1...D.
CR es un parámetro que controla la tasa de cruce. Nótese que la comparación se hace componente por componente, por lo que el vector de prueba obtenido será una mezcla de los vectores mutados y el original, aumentándose la diversidad de la población.

Selección

Finalmente, la selección se realiza simplemente comparando los vectores de prueba con los originales, de manera que el vector de la generación siguiente será aquel que tenga el mejor valor de función de coste:

$$x_i^{k+1} = \begin{cases} t_i^k; & fit(t_i^k) < fit(x_i^k) \\ x_i^k; & otro\ caso \end{cases}$$

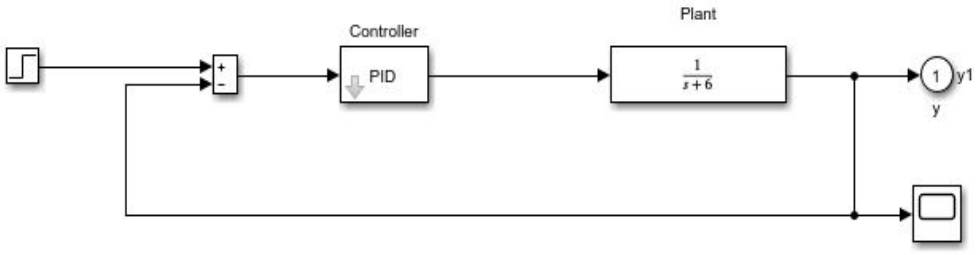
La población evolucionará hacia los individuos con un mejor valor asociado para la función de coste lo que quiere decir que devolverá la mejor solución después de un número dado de iteraciones.

Ejemplo

En la carpeta Ejemplo de esta practica se encuentran 4 ficheros:

- *tf2.slx*
- *Pid_DIff_Evol_3.m*
- *tracklsq3.m*
- *devec3.m*

El fichero ***tf2.slx*** contiene el esquema de control basada en un PID para la función de transferencia propuesta para controlar:



La función ***Pid_DIff_Evol_3.m*** (de la carpeta Ejemplo de esta práctica) minimiza una función de coste definida por el usuario para una población determinada utilizando el algoritmo DE de Rainer Storn (http://www.icsi.berkeley.edu/~storn/code.html); es decir, hace evolucionar una población inicial hacia los elementos que tienen un valor coste menor.

Argumentos de salida:

- bestmem: [Kp Ki Kd].
- error: valor de coste para la mejor solución.
- population: población, el primer elemento de cada fila es el coste asociado del elemento de esa fila.
- CONV: evolución de la función de error con las iteraciones.
- M: función de transferencia del sistema con el PID calculado, en bucle cerrado.
- Pm: margen de fase.
- Wpm: frecuencia de margen de fase.

Parámetros que se pueden ajustar

- Variables del DE:
  - iter_max= .... [Número de iteraciones]
  - NP= ... [Tamaño de la población]
  - MAX_K_VAR= ...[Máximo valor inicial para las constantes del PID]
  - F=... [Factor de amplitud de diferencias del DE, 0-2]
  - CR=... [Constante de cruce, 0-1]

Los argumentos elegidos son básicos para tener éxito. El algoritmo funcionará bien sólo si el intervalo elegido cubre el mínimo global. El éxito también dependerá del valor elegido para la constante F. Una buena aproximación inicial es elegir un valor en el intervalo [0.5 1]. Por ejemplo, 0.8. La constante de cruce CR ayuda a mantener la diversidad de la población. El tamaño de la población (NP) es importante pero no es crítico para este problema. Una buena elección inicial es 10*D, siendo D el número de cromosomas (3 en este caso). Dependiendo de la complejidad de problema se elegirá un tamaño adecuado.

En esta práctica se utilizará está función para calcular los parámetros de un PID que controle el motor de corriente continua en velocidad. Estos parámetros son Kp, Ki y Kd.

El algoritmo evolucionará hacia la combinación de estos tres parámetros que minimiza la función objetivo. La función objetivo viene dada por ***tracklsq3.m***. Representa la diferencia entre la entrada y la salida; Esta función hace una simulación del motor con los parámetros del PID y obtiene la salida de la planta.

La función de coste evalúa las diferencias entre la salida y la entrada y evoluciona hacia el elemento con una mejor función de coste, es decir, el PID que mejor controla la planta.

Trabajo para realizar:

1) Diseño del PID mediante el algoritmo genético Differential Evolution (DE).

   En este paso se obtienen los parámetros del controlador PID utilizando algoritmos genéticos. Los archivos necesarios están en el directorio Practica1/Ejemplo de este curso.

   1) Una vez estamos situados desde Matlab en la carpeta de la práctica (Practica1/Ejemplo), abrir el archivo *Pid_DIff_Evol_3.m.* Mediante la modificación de ese archivo (los ajustes de variables del DE) se obtendrán los parámetros del PID que optimizan la respuesta del sistema.

   2) Los parámetros a modificar son los siguientes:
      a. Función de transferencia de la planta, se puede modificar en el fichero *tf2.slx*
      b. Variables del DE (se modifican en el fichero *Pid_DIff_Evol_3.m*)
         i. iter_max: número de iteraciones
         ii. NP: tamaño de la población.
         iii. MAX_K_VAR: amplitud de la zona de búsqueda inicial
         iv. F: factor de amplitud diferencial del DE, 0-2.
         v. CR: constante de cruce, 0-1.
   3) Una vez fijados los parámetros anteriores, lanzar el algoritmo. Para ello, ejecutamos el fichero *Pid_DIff_Evol_3.m.*

      La función de coste tratará de optimizar la respuesta ante entrada escalón del sistema. El objetivo de la práctica será fijar unos parámetros adecuados para las variables propuestas en el anterior apartado de tal forma que la respuesta se comporte de la mejor forma posible.

      La solución vendrá dada por la variable *x.* El PID calculado por el optimizador es el siguiente:

      x: [Kp Ki Kd]

   4) Comprobación de resultados. Abrir el archivo ***tf2.slx*** y ejecutar el programa (el programa completa por defecto el controlador PID con los parámetros obtenidos).

   5) Finalmente, se pide anotar todos los valores y gráficas obtenidas durante la práctica, y realizar un guión en el que se cuente brevemente lo que se ha hecho y los resultados obtenidos de forma analítica.

   6) Repetir el mismo procedimiento para las siguientes funciones de transferencias, analizando los resultados:

      a) $G(s) = \frac{1}{0.51s^2+s+1}$

      b) $G(s) = \frac{1}{0.51s^2+s}$