**Control Inteligente 1- Informe control con algoritmos geneticos**

**Por: Cristian Alejandro Agudelo Zapata**

**Codigo utilizado**

|  |
| --- |
| # -\*- coding: utf-8 -\*-  """  Created on Thu Feb 3 09:33:26 2022  @author: CRISTIAN  """  import control  from control.matlab import \*  import matplotlib.pyplot as plt  import numpy as np  from min\_max\_con\_libreria import \*  T=np.arange(0,10,0.01)  num=[1]  den=[1,2,2]  sys=tf(num,den)  #creacion de elementos de graficacion  fig,(ax1,ax2,ax3)= plt.subplots(1,3)  fig.suptitle('Control con algoritmo genetico')  #creo el escalon  escalon=np.array([0])  t=len(T)  escalon=np.append(escalon,np.ones(1000))  #grafico la ft en lazo abierto y lazo cerrado ante un escalon unitario  LA, T =step(sys,T)  ax1.set\_title('LA')  ax1.plot(T,LA)  dataLA=stepinfo(sys)  tssLA=dataLA['SettlingTime']  tff=control.feedback(sys, sys2=1, sign=- 1)  LC, T = step(tff,T)  ax2.set\_title('LC')  ax2.plot(T,LC,'y')  dataLC=stepinfo(tff)  Mp=dataLC['Overshoot']  tssLC=dataLA['SettlingTime']  #print('Respuesta temporal en LA:')  #inf1=control.damp(sys)  #print('')  #print('Respuesta temporal en LC:')  #inf2=control.damp(tff)    #Funcion objetivo para control PID  def funcion\_objetivo(kp,ti,td):  #Defino la funcion de transferencia en este caso es de segundo orden  num=[1]  den=[1,2,2]  sys=tf(num,den)    #Defino el controlador  numc=[kp\*ti\*td,kp\*ti,kp]  denc=[ti,0]  sysc=tf(numc,denc)  ftotal=series(sys,sysc)  syst=control.feedback(ftotal,sign=- 1)  #inf3=control.damp(syst)  try:  #algunos parametros de respuesta en el tiempo (Mp,TSS,Eee)  #zita1=np.array([])  #zita1=inf2[1]  # pi=3.141592653589793  # Mp=np.exp((-pi\*zita1[1])/(np.sqrt(1-zita1[1]\*zita1[1])))  # wn1=np.array([])  # wn1=inf2[0]  # Tss=4/(zita1[1]\*wn1[0])  Tss=tssLA\*0.75 if dataLC['SettlingTime']<0.75\*tssLA else dataLC['SettlingTime']  Mpp=10 if dataLC['Overshoot']<10 else dataLC['Overshoot']  Eee=1-LC[-1]  fitness=abs(10-Mp)+Tss+Eee  #fitness=abs(10-Mp)+(((0.75\*tssLA-Tss)/Tss)+(Eee)  except:  fitness=1  return fitness  poblacion = Poblacion(  n\_individuos = 15,  n\_variables = 3,  limites\_inf =[0.001,0.001,0.001],  limites\_sup =[100,1,1],  verbose = False  )  poblacion.optimizar(    funcion\_objetivo =funcion\_objetivo,  optimizacion = "minimizar",  n\_generaciones = 100,  metodo\_seleccion = "rank",  elitismo = 0.2,  prob\_mut = 0.1,  distribucion = "uniforme",  media\_distribucion = 1,  sd\_distribucion = 1,  min\_distribucion = -1,  max\_distribucion = 1,  parada\_temprana = True,  rondas\_parada = 10,  tolerancia\_parada = 10\*\*-16,  verbose = False  )  nume=[poblacion.valor\_variables\_optimo[0]\*poblacion.valor\_variables\_optimo[1]\*poblacion.valor\_variables\_optimo[2],poblacion.valor\_variables\_optimo[0]\*poblacion.valor\_variables\_optimo[1],poblacion.valor\_variables\_optimo[0]]  dene=[poblacion.valor\_variables\_optimo[1],0]  syse=tf(nume,dene)  ftotall=series(sys,syse)  T=np.arange(0,10,0.01)  syst=control.feedback(ftotall,sign=- 1)  C\_eval, T = step(syst,T)  ax3.set\_title('Control optimizado')  ax3.plot(T,C\_eval,'r')  dataC=stepinfo(syst) |

Los mejores resultados obtenidos de acuerdo al algoritmo anterior fueron los siguientes:

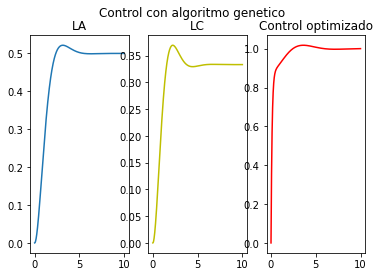


Figura 1

En la figura 1 se observa las funciones correspondientes en lazo abierto y lazo cerrado además de la implementación del controlador PID por medio de la búsqueda de los parámetros Kp,Ti y Td para que el controlador elimine el error en estado estable primero que todo, ya después se analiza lo que es el sobre impulsó, el tiempo de establecimiento y el esfuerzo de la ley de control.

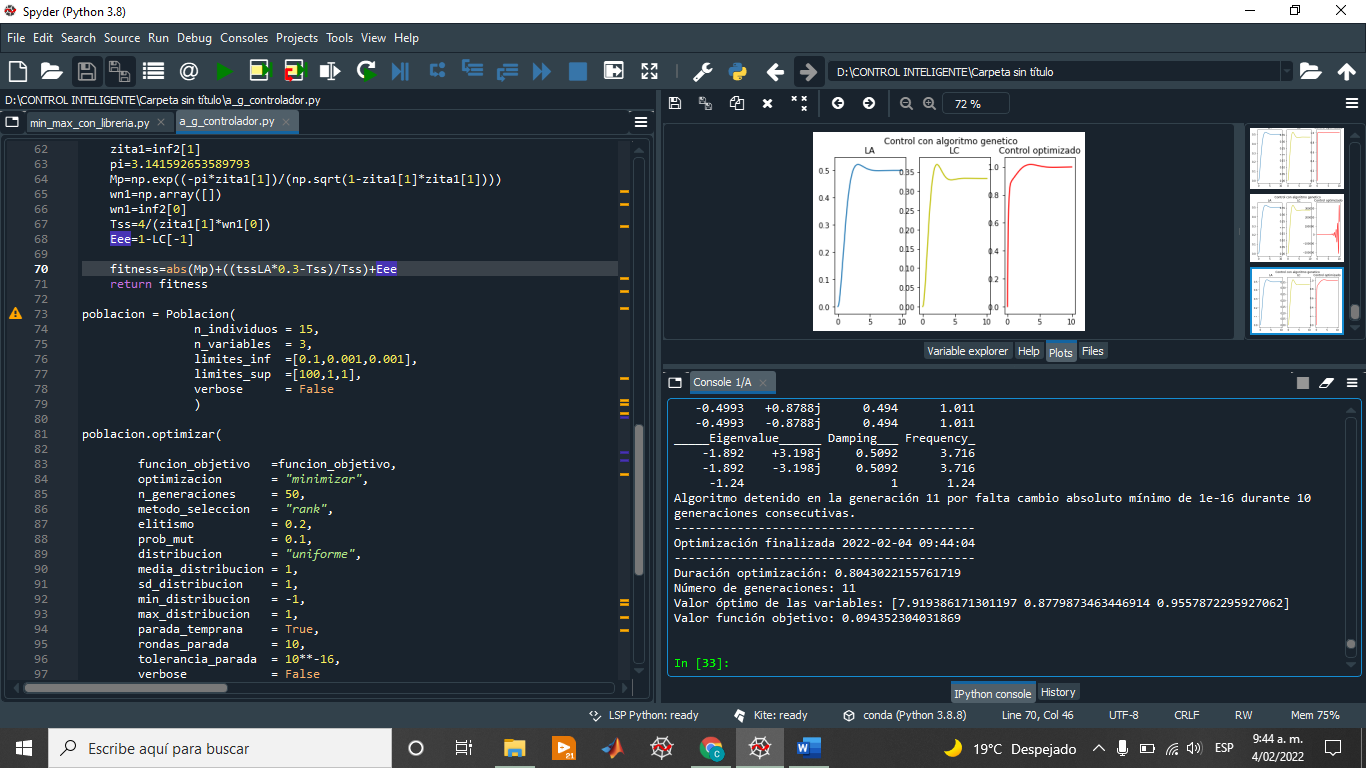


Figura 2

En la figura 2 se observa el valor óptimo de las variables siendo el primer valor del vector Kp, el segundo Ti y el tercero Td. Este controlador en su función objetivo en este el de minimizar el Mp, tss y el Eee, obtuvo muy bajo valor siendo lo que se necesita en el controlador, pero la ley de control al principio de la respuesta del sistema se puede comportar muy agresivamente,

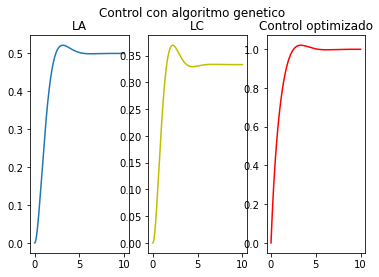


Figura 3

En la figura 3 se observa igualmente las 3 salidas del sistema siendo la de la derecha el control optimizado.

Los valores encontrados por el algoritmo genético se presentan en la figura 4, en la que el valor de la función objetivo es mayor respecto a la mostrada anteriormente en la figura 2, sin embargo, la respuesta del sistema se obtiene mas suave en los primeros instantes de tiempo comparando con la figura 1. Este controlador además cumple con los requisitos establecidos en la función fitness, en la que se le pidió al sistema en estabilizar en un valor aproximado al 75% del tiempo de establecimiento en lazo abierto, además de que elimina el error en estado estable y su sobre impulsó no supera el 10% que es lo que se buscó.

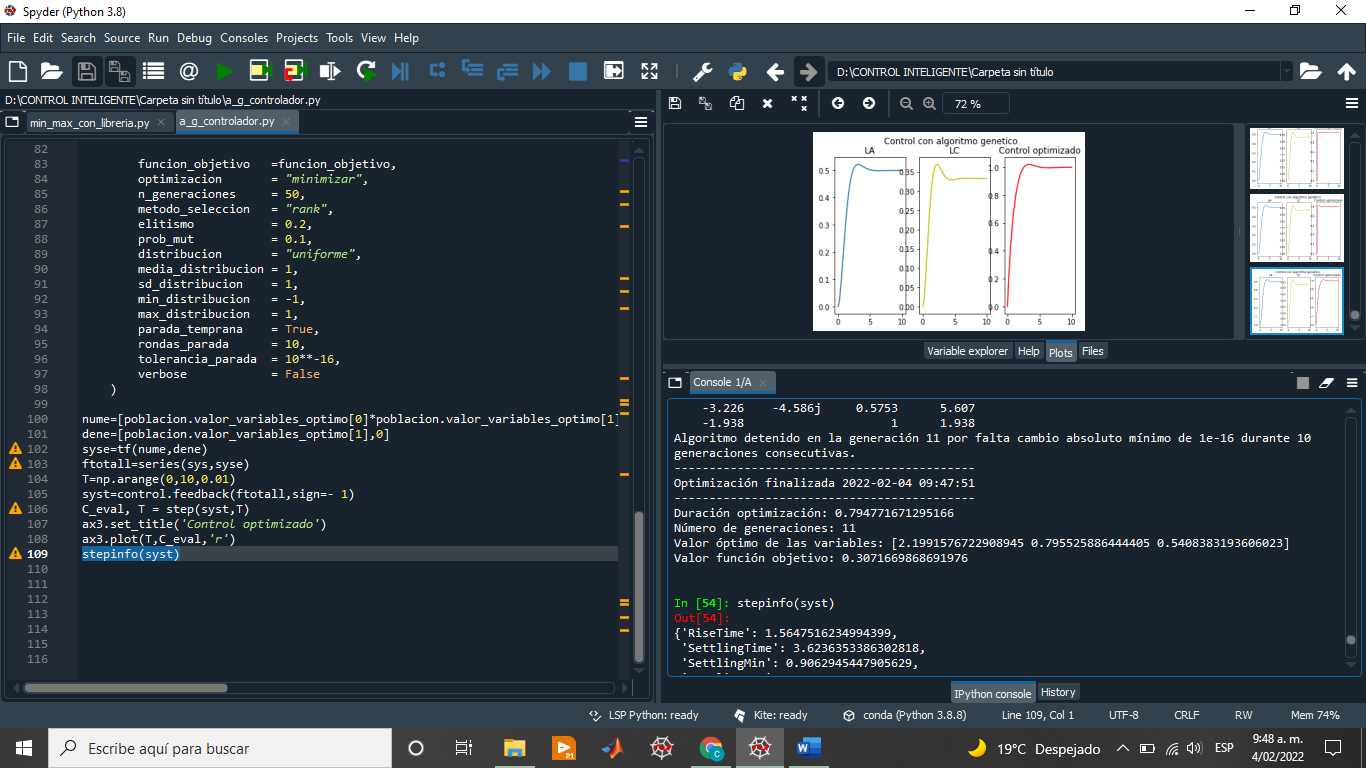


Figura 4

En la figura 5, se presentan las metricas de desempeño expuestas anteriormente en la que se cumple con lo indicado en la funcion fitnnes.

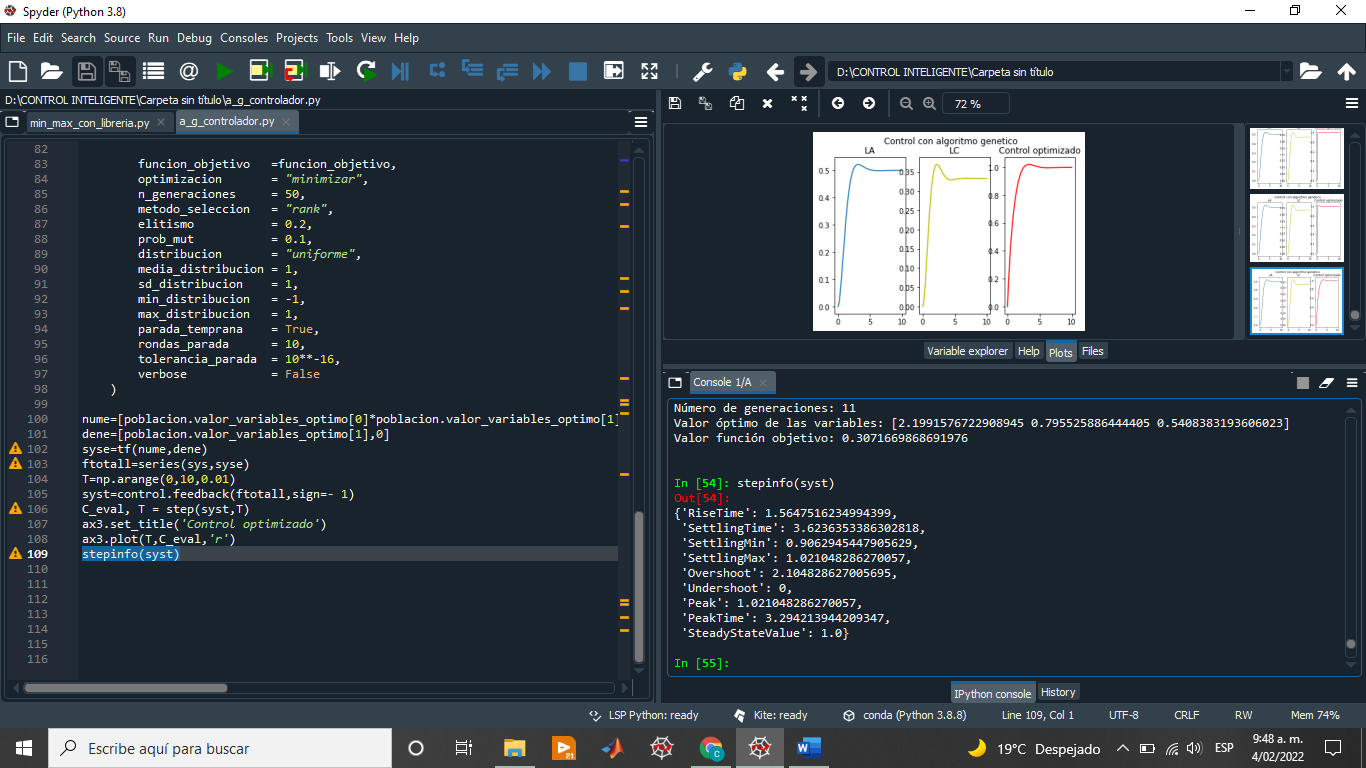


Figura 5

La respuesta de este controlador observado en la figura 6, presenta más sobre impulsó que los controladores analizados anteriormente, además de que tiene mas oscilaciones en el tiempo aumentando así el valor de la función objetivo.

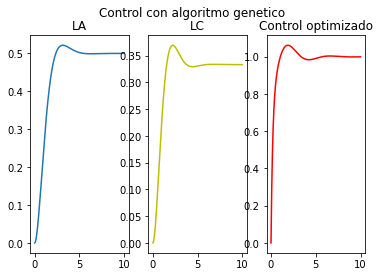


Figura 6

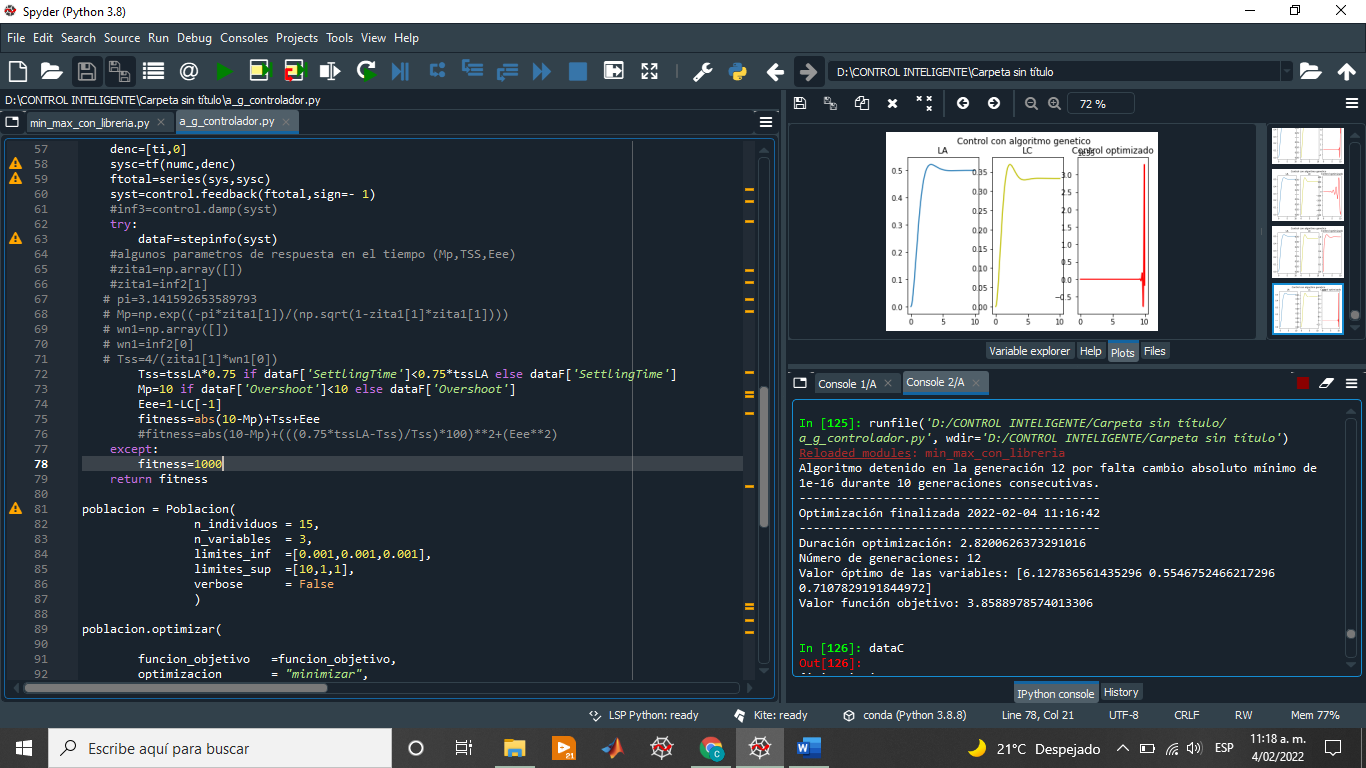


Figura 7

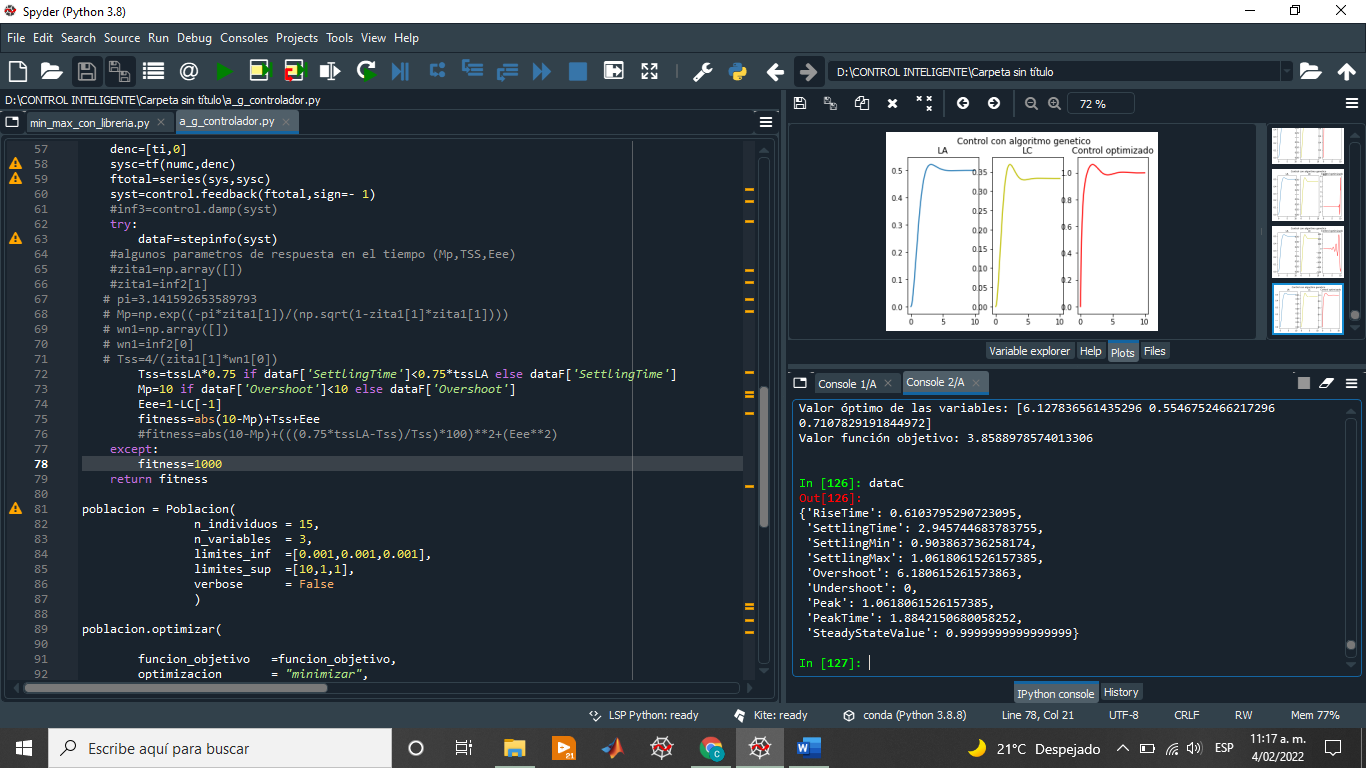


Figura 8

En la figura 9 se observa los parámetros iniciales de la planta en lazo abierto, con los que se analiza los resultados deseados del controlador frente a la función fitness planteada en cada uno de ellos

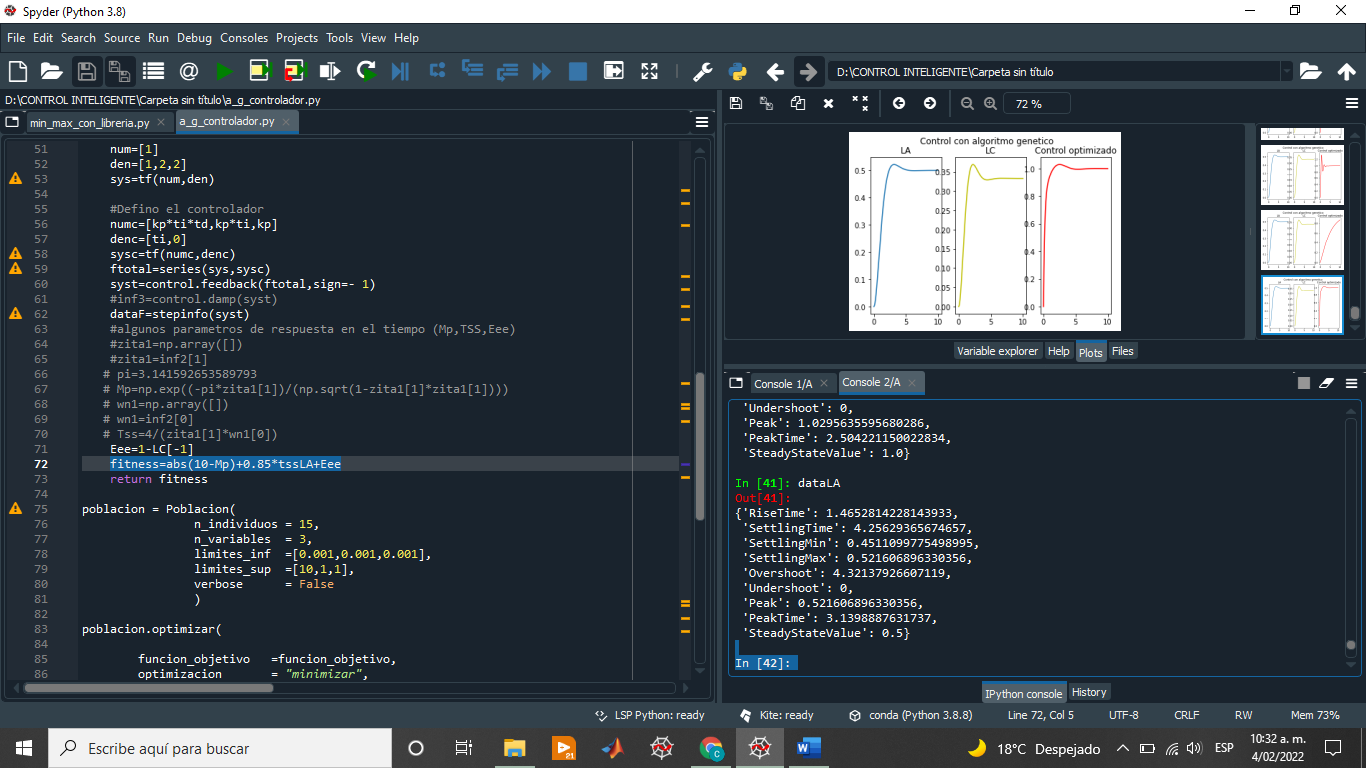


Figura 9

En la figura 10 se observa igualmente la respuesta del controlador con otros parametros, en que se observa que cumple con el tiempo de establecimiento igualmente, el sobreimpulso no sobrepasa el 10% y el error de estado estable es igual 0, ademas de que al principio la planta no responde agresivamente dandole cuidado a la ley de control, sin embargo la funcion fitness de este controlador no es mas minima que la se muestra en la figura 4.

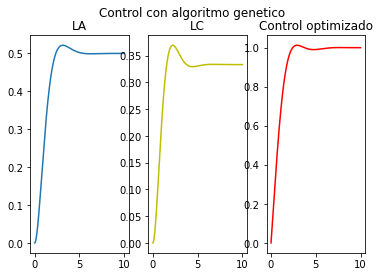


Figura 10

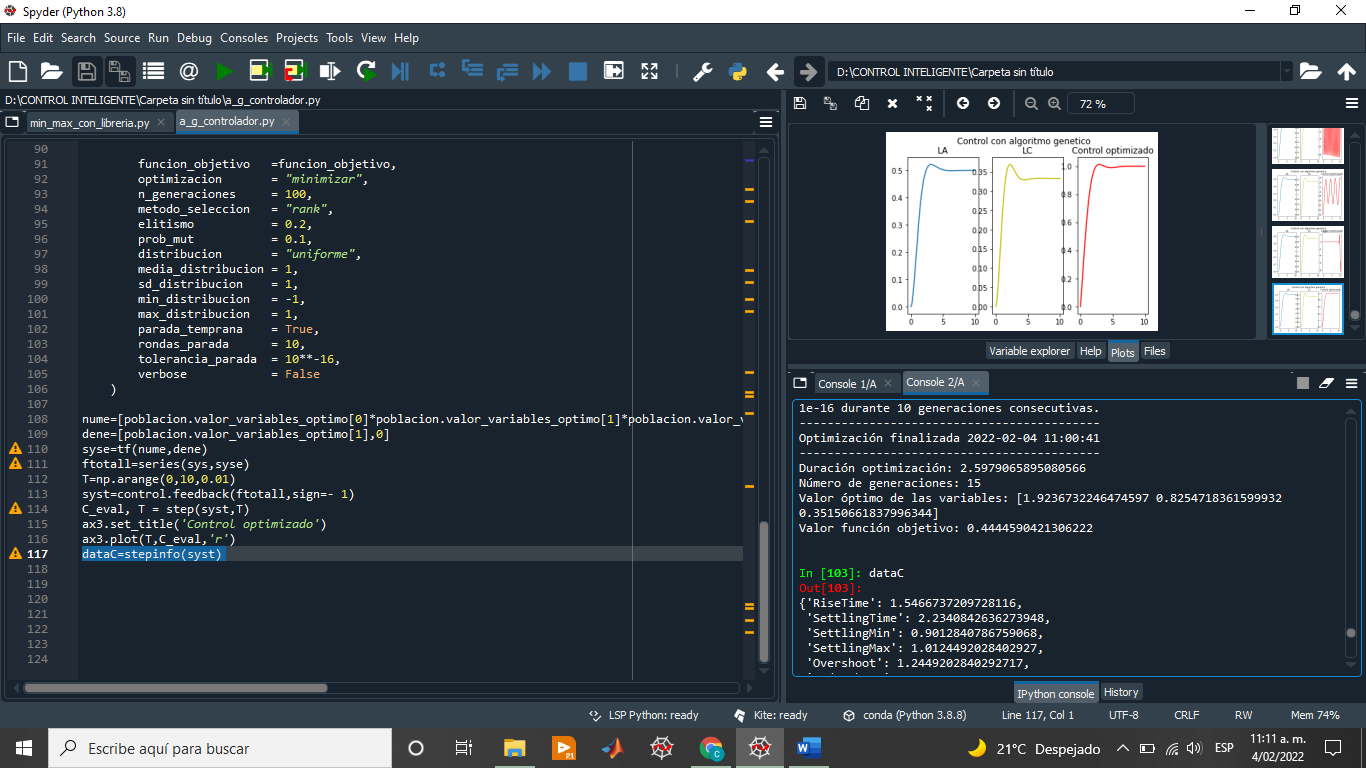


Figura 11

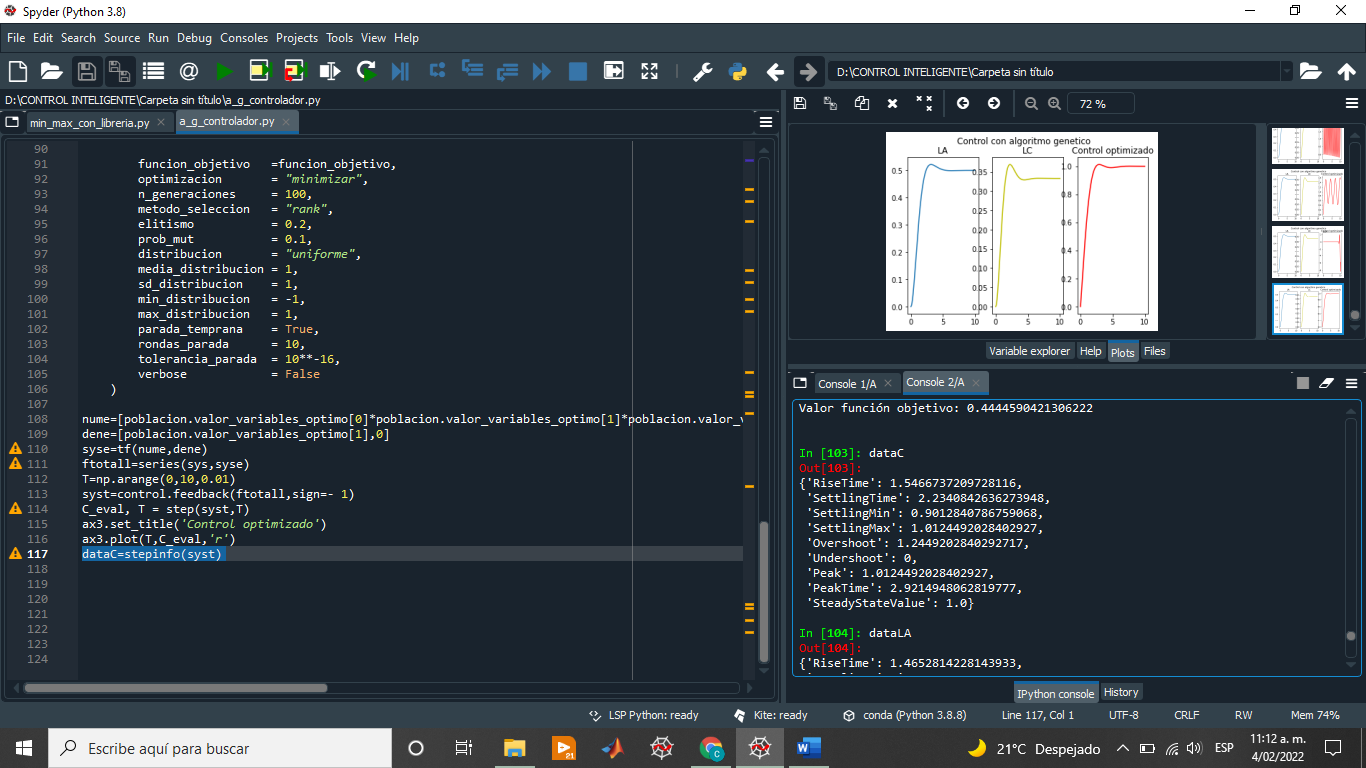


Figura 12

**CONCLUSIONES**

* La búsqueda de los parámetros de un controlador PID mediante los algoritmos genéticos se cumple satisfactoriamente, quedando únicamente por analizar los resultados obtenidos y los deseados.
* Al aumentar la población de los individuos tenemos más probabilidad de tener un valor de la función objetivo menor, ya que hay más probabilidad de que se escojan individuos con mejores características y estos tienen más probabilidad de pasar a las siguientes generaciones.
* Haciendo las pruebas correspondientes, observamos que el controlador cumple con las especificaciones dadas de sobre impulso y de tiempo de establecimiento que se indicó. Si se analiza la respuesta de todos los controladores y se cuida la ley de control, los parámetros obtenidos en la figura 4 son los más óptimos.
* Se observa que, para esta planta y controlador, si el valor de Kp aumenta este es proporcional a la velocidad con que arranca la planta con el controlador.