Scientific Journal of Impact Factor (SJIF): 5.71
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406
International Journal of Advance Engineering and Research Development
International Conference of Trends in Information, Management, Engineering and Sciences (ICTIMES)
Volume 5, Special Issue 02, Feb.-2018 (UGC Approved)

Genetic Algorithm based Fractional Order PID Control for Temperature Control Plant

Dr.G.Sridhar

Professor, Department of Electronics and Communication Engineering,
Malla Reddy College of Engineering

Abstract- *With the continuous innovation and to control the temperature produced by appliances to help the energy consumption. The energy demand and control motor temperature, speed, positions along with the congested transmission systems, fractional order PID[1] is suggested as the best solution So many methods are involved to control temperature but sometimes they fail because of increased temperature. Temperature sensors are used many appliances but due to this failure sometimes accidents happen. To overcome this issue we introduced fractional order PID controller to control the temperature [2] before attending the cutoff level. To design the fractional order PID controller[1] using a genetic algorithm to get better value for the fractional orders of the corresponding systems.*

Key words: *FOPID, GA Tuning, Temperature control system*

INTRODUCITON

The FO-PID[1] control has a fractional integral and a differential elements in which these orders are non-integral. Generally, as the physical plant has a fractional characteristic, it is expected that the fractional controller will be effective for actual plants. There are some advantages of fractional control scheme, it was reported that $PI^{\alpha}D$ control system has a robust characteristics for the input saturation .Implementation of FOPID[1] finite order approximation is required, fractional elements have infinite order. There have been various researches for approximation of fractional elements by the finite order filter. The SMP (short memory principle) method is effective in terms of implementation and approximation accuracy. The SMP method gives the discrete approximation of the fractional element and provides the better approximation accuracy than other digital methods. The binomial coefficients at the beginning were reduced as time advances. The integral and differential are approximated using the data during recent interval. The output error remains in steady state as the FOPID approximated by SMP. To eliminate the steady state error, divide the fractional integral into traditional integral s^{-1} , it is called distributed implementation.

1. *The implementation method of fractional order integration, which has the integration characteristics in low frequency is examined.*
2. *Approximation accuracy using SMP is evaluated.*

TYPES OF FRACTIONAL ORDER CONTROLLER:-

Historically there are four major types of fractional order controllers[1],
- CRONE Controller.
- Tilted Proportional and Integral(TID) Controller.
- Fractional Order PID (FOPID) Controller.
- Fractional Order Lead-Lag Compensator.

REASON FOR SELECTING FOPID:-

Most of the physical plant has a fractional characteristic, it's expected that the fractional controller will be effective for actual plants.

BLOCK DIAGRAM:-

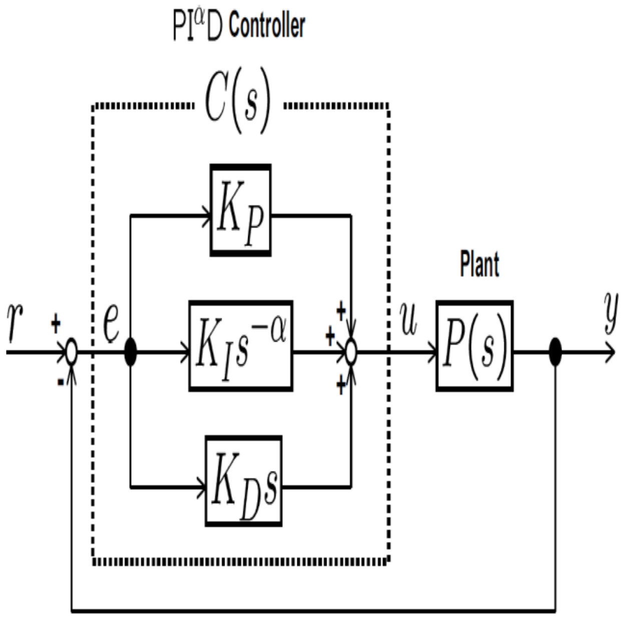


Fig. 1 BLOCK DIAGRAM OF PI$^{\alpha}$D CONTROLLER

PI$^{\alpha}$D controller one of the FOPID controllers with fractional integrator. In case α=1, the Controller is equivalent to the traditional PID controller. Advantages of the fractional Control scheme, it was reported that PI$^{\alpha}$D control system has robust characteristics for the saturation of the input.
Fractional order PID controller output is

$C_f(s) = K_p + K_i / s^{\lambda} + K_d s^{\mu}$

The FOPID [1]controller needs to design five parameters, K_p, K_I, K_d, λ, μ. The order λ and μ are not necessary to be integers , they will be any real numbers.

- Selecting λ=1, μ=1, a classical PID can be obtained.
- Selecting λ=1, μ=0, a PI$^{\lambda}$ can be obtained.
- Selecting λ=0, μ=1, a PD$^{\mu}$ can be obtained.
- Selecting λ=0, μ=0, a gain can be obtained.

The PI$^{\lambda}$ D$^{\mu}$-controller is more flexible and gives an opportunity to better adjust the dynamical properties of a fractional-order control system.

system transfer function [2]

P(s) = ((1)/(14994s^1.98+6009.5s^0.97+1.69)

FOPID α-APPROXIMATION TECHNIQUE:

A commonly used Three definition of the fractional calculus is the Riemann-Liouvilledefinition[7],

$$aD_t^{\alpha} f(t) = \frac{1}{\tau(m-\alpha)} \left(\frac{d}{dt}\right)^m \left(\int_0^{\tau} \frac{f(\tau)}{(t-\tau)^{(1-(m-\alpha))}} d\tau \right)$$

$$ {}_aD_t^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)^m \left(\int_a^t \frac{f(\tau)}{(t-\tau)^{1-(m-\alpha)}}\mathrm{d}\tau\right) $$

An alternative definition, based on the concept of fractional differentiation, is the Grunwald-Letnikov definition given by [7],

$$ {}_aD_t^\alpha f(t) = \lim_{h\to 0}\frac{1}{\Gamma(\alpha)h^\alpha}\sum_{k=0}^{(t-\alpha)/h}\frac{\Gamma(\alpha+k)}{\Gamma(k+1)} f(t-kh) $$

Caputo fractional derivatives,

$$ {}_aD_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)}\int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}}d\tau, \quad \text{for } n-1<\alpha<n. $$

Most frequently referred definition, grunwald-letnikov definition. $S^{-\alpha}$denotes the fractional integration operator.The operator means the integral in case that $\alpha< 0$ and the differential in case that $\alpha> 0$.
Oldham and Spanier (1974)[6]

$$ \frac{d^q f(\beta x)}{dx^q} = \beta^q \frac{d^q f(\beta x)}{d(\beta x)^q} $$

K.S. Miller and B. Ross (1993)[6].

$$ D^\alpha f(t) = D^{\alpha_1} D^{\alpha_2} \ldots D^{\alpha_n} f(t) $$
$$ \alpha = \alpha_1 + \alpha_2 + \ldots + \alpha_n $$
$$ \alpha_i < 1 $$

Kolwankar and Gangal (1994) [6].

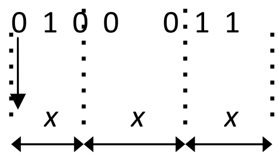
$$ \mathbf{D}^q f(y) = \lim_{x\to y}\frac{d^q\left(f(x)-f(y)\right)}{d(x-y)^q} $$

GENETIC ALGORITHMS (GA):

Most real world optimization problems involve complexities like discrete, continuous or mixed variables, multiple conflicting objectives, non-linearity, discontinuity and non-convex region. The search space (design space) may be so large that global optimum cannot be found in a reasonable time. The existing linear or nonlinear strategies might not be economical orcomputationally cheap for resolution such issues. numerous random search strategies likesimulated tem pering, organic process algorithms (EA) or hill rising isutilized in such things. EAs have the advantage of being applicable to anycombination of complexities(multi-objective,non-linearity etc) and can also be combined with any existing native search or alternative strategies. Numerous techniques that create use of Ea approach area unit Genetic Algorithms (GA), organic process programming, evolution strategy, learning classifier system etc. of these Ea techniques operate primarily on a population search basis. during this lecture Genetic Algorithms, the foremost in style Ea technique, is explained

Concept

EAs start from a *population* of possible solutions (*called individuals*) and move towards the optimal one by applying the principle of Darwinian evolution theory i.e., *survival of the fittest*. Objects forming possible solution sets to the original problem is called *phenotype* and the encoding (representation) of the individuals in the EA is called *genotype*. The mapping of phenotype to genotype differs in each EA technique. In GA which is the most popular EA, the variables are represented as strings of numbers (normally binary). If each design variable is given a string of length '*l*', and there are *n*such variables, then the design vector will have a total string length of *'nl'.*



An individual consists a genotype and a fitness function. *Fitness* represents the quality of the solution (normally called *fitness function*). It forms the basis for selecting the individuals and thereby facilitates improvements.A flowchart indicating the steps of a simple genetic algorithm is shown below.

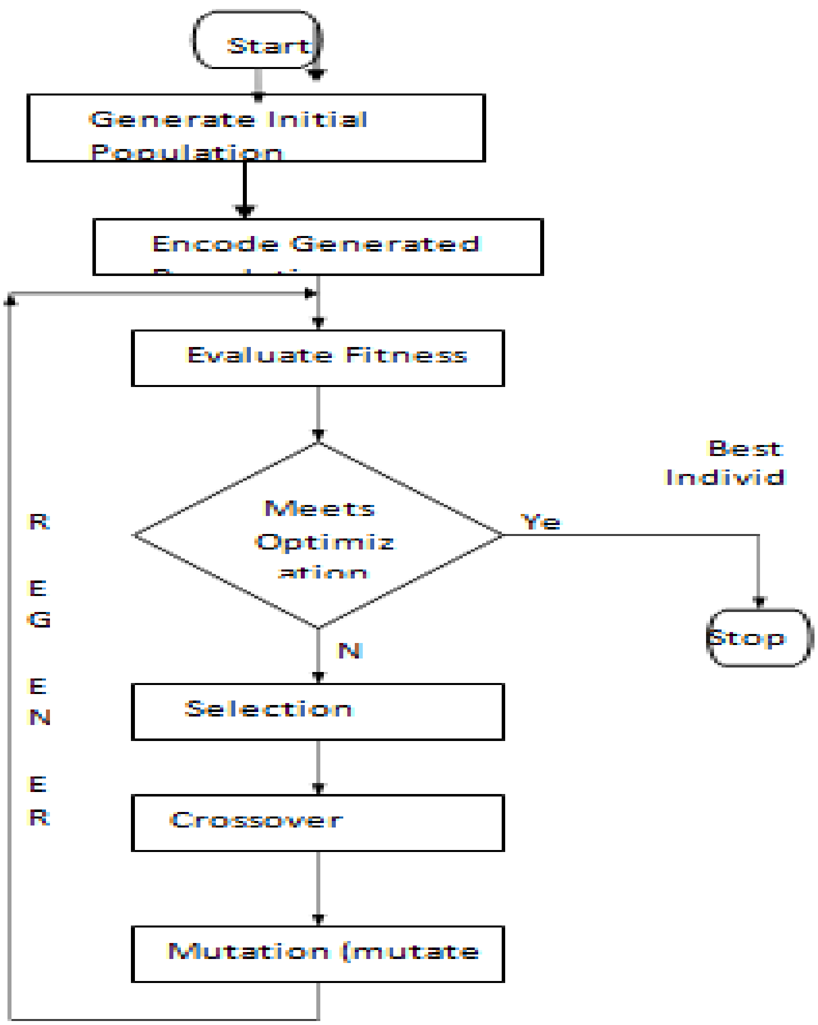


Fig 2 Flow chart for GA tuning algorithm

The pseudo code for a simple EA is given below

i = 0
initialize population p0
Evaluate initial population
While (! Terminate condition)
{i = i+1Perform competitive selection Create population P_i from P_{i-1} by recombination and mutation Evaluate population P_i}

The initial population is usually generated randomly in all EAs. The termination condition may be a desired fitness function, maximum number of generations etc. In selection, individuals with better fitness functions from generation '*i*' are taken to generate individuals of '*i+1*'th generation. New population (*offspring*) is created by applying *recombination* and *mutation* to the selected individuals (*parents*). Recombination creates one or two new individuals by swaping (crossing over) the genome of a *parent* with another. Recombined vidualis then *mutated* by changing a single element (genome) to create a new individual.

Finally, the new population is evaluated and the process is repeated. Each step is described in more detail below.

Parent Selection

After fitness function evaluation, individuals are distinguished based on their quality. According to Darwin's evolution theory the best ones should survive and create new offspring for the next generation. There are many methods to select the best chromosomes, for example roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and others. Two of these are briefly described, namely, roulette wheel selection and rank selection:

Roulette Wheel Selection:

Parents square measure chosen per their fitness i.e., every individual is chosen with a likelihood proportional to its fitness price. In alternative words, betting on the share contribution to the full population fitness, string is chosen for conjugation to create successive generation. This way, weak solutions square measure eliminated and robustsolutions survive to create successive generation. for instance, contemplate a population containing four strings shown within the table below. every string is made by concatenating four substrings that represents variables a,b,c and d. Length of every string is taken as four bits. the primary column represents the potential answer in binary kind. The second column provides the fitness values of the decoded strings. The third column provides the share contribution of every string to the full fitness of the population. Then by "Roulette Wheel" technique, the likelihood of candidate onebeing chosen as a parent of successive generation is twenty eight.09%. Similarly, the likelihood that the candidates a pair of, 3, four are chosenfor successive generation squaremeasure nineteen.59,12.89and39.43 severally. These possibilities square measure diagrammatical on a chart, so four numbers square measure haphazardly generated between one and a hundred. Then, the probability that the numbers generated would fall within the region of candidate a pair of may be once, whereas for candidate four it would be double and candidate one over once and for candidate three it should not fall in the least. Thus, the strings square measure chosen to create the oldsters of successive generation.

Rank Selection:

The previous style of choice might have issues once the fitnesses disagree noticeably. for instance, if the most effective body fitness is ninetieth of the complete game equipment then the opposite chromosomes can have only a few probabilities to be chosen. Rank choice 1st ranks the population so each body receives fitness from this ranking. The worst can have fitness one, second worst two etc. and therefore the best can have fitness N (number of chromosomes in population). By this, all the chromosomes can have an opportunity to be chosen. however this methodology will cause slower convergence, as a result of the most effective chromosomes might not disagree a lot of from the others.

Crossover

Selection alone cannot introduce any new people into the population, i.e., it cannot realize new points within the search area. These are generated by genetically-inspired operators, of that the foremost renowned are crossover and mutation.

Crossover is of either one-point or two-point theme. In one purpose crossover, selected try of strings is cut at some random position and their segments ar swapped to make new try of strings. In two-point theme, there'll be 2 break points within the strings that ar at random chosen. At the break-point, the segments of the 2 strings ar swapped so
new set of strings are formed. For example, let us consider two 8-bit strings given by '10011101' and '10101011'.
Then according to one-point crossover, if a random crossover point is chosen after 3 bits from left and segments are cut as shown below:

100 | 11101
101 | 01011

Crossover is not usually applied to all pairs of individuals selected for mating. A random choice is made, where the probability of crossover being applied is typically between 0.6 and 0.9.

Mutation

Mutation is applied to each child individually after crossover. It randomly alters each gene with a small probability (generally not greater than 0.01). It injects a new genetic character into the hromosome by changing at random a bit in a string depending on the probability of mutation. Optimization Methods: Advanced Topics in Optimization – Evolutionary Algorithms for 7 Optimization and Search
Is mutated as 10111**1**11

It is seen in the above example that the sixth bit '0' is changed to '1'. Thus, in mutation process, bits are changed from '1' to '0' or '0' to '1' at the randomly chosen position of randomly selected strings.

Real-coded GAs

As explained earlier, GAs work with a coding of variables i.e., with a discrete search space. GAs have also been developed to work directly with continuous variables. In these cases, binary strings are not used. Instead, the variables are directly used. After the creation of population of random variables, a reproduction operator can be used to select good strings in the population.

Results

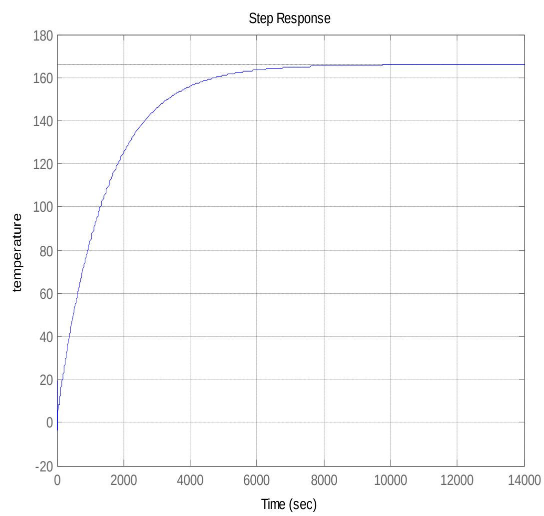


Fig 3 ZN FOPID based temperature system

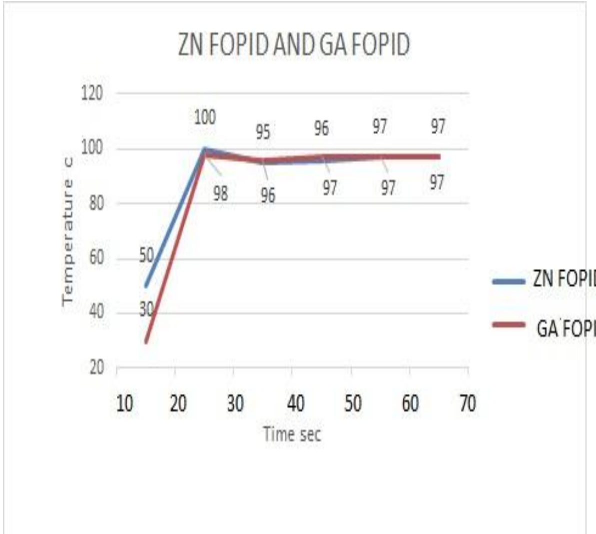


Fig 4 ZN FOPID and GAFOPIDbased temperature system

Conclusion:

Design and Implementation method of the fractional order PID controller for Fractional order process[2] in MATLAB simulation has been done and results are compared with ZNFOPID[1] controller. The settling time, rise time of the GAFOPID controller are better compared to that of ZNFOPID controller. In future try to find the better solution using new algorithms.

References

1. G.sridhar"Design and Implementation of fractional order PID controller" CIIT(2012)
2. Dr.G.sridhar "Complex Circuit Model Analysis Using CDTA and SNAP" International Journal of Applied Engineering Research(2015)
3. Nobutomo Matsunga, Koji Sasano and Hiroshi Okajima: "AN IMPLEMENTATION OF FRACTIONAL ORDER PID CONTROLLER WITH DYNAMIC QUANTIZER CONSIDERING THE MEMORY CONSTRAINT"IEEE International conference on control Applications (2010).
4. I.Petras and B.M.vinagre "Practical application of digital fractional-order control to temperature control". Actamontanstica slovaca,7-2,pp.131-137(2002)
5. R.Schreier andG.C.Temes: Understanding Delta-Sigma Data converters, Wiley-IEEE press(2004).
6. Shantanu Das: "Functional fractional calculus for system identification and controls", springer-verlag(2007).

About author

G.Sridhar received the Ph**.d**degree in electronics and instrumentation engineering from the SR university,India, in 2015, the M.E**.** degree in Control and Instrumentation from Anna University, Chennai, India, in 2012. He is currently working as a Professor in Department of ECE, Malla Reddy College of Engineering, Hyderabad. His research interests include communication and networks,wireless communication networks, Control systems, Fuzzy logic and Networks, AI, Sensor Networks.