Temperature control for a chemical reactor using a new Genetic Algorithm

Fayez Areed#1, Mostafa A. El-Hosseini *2, M S Saraya#3

#Computer and control systems Engineering Department, Mansoura University
Egypt - Mansoura
mohamedsabry83@mans.edu.eg

Abstract—Genetic algorithms (GAs) have been fairly successful in a diverse range of optimization problems, providing an efficient and robust way for guiding a search even in a complex system and in the absence of domain knowledge. This paper, presents a comparative study for a chemical reactor system by conventional P, PI, PID, and by using genetic algorithm for the same plant. The results obtained here, assure the actual possibility of using GA to identify and controlling plants. The major efforts are to adjust the controller in order to minimize the steady state error. GA offer, an alternative approach both for identification and control of nonlinear processes in process engineering.

Keywords—genetic algorithm, GA, PID Controllers, Stochastic systems, chemical reactor, optimization.

I. INTRODUCTION

In the last two decades, the growth in interest in heuristic search methods for optimization has been quite dramatic. Heuristics have now attained considerable respect and are extremely popular in the field of optimization. One of the most interesting developments is in the application of genetic algorithms (GAs), which has regularly been featured, in control engineering. The genetic algorithm is one of the newly developed field and one of the important topics in research of computational intelligence. Application of genetic algorithms, to research of computer architecture is not new. The algorithm is used for research of Very Large Scale Integrated Chip (VLSI) design to find the optimized area and optimized number of VIAs for the situation [1] – [5]. These references and [6] give more details for genetic algorithm, which we do not go over deeply. In this paper, we only provide the minimum knowledge to understand operations of the genetic algorithm.

I.1 General Genetic Algorithm

Genetic algorithms are rich in application across a large and growing number of disciplines. Really genetic algorithm supports computer programming. This research provides an Introduction to Genetic Algorithms and its components. A GA is a stochastic optimization method based on the biological principles of Darwinian evolution [7]. GA incorporate operators that mimic natural selection and reproduction (on a simplistic level) using a probabilistic search on a population of designs. The population ‘evolves’ through the application of genetic operators to determine the design that is best adapted to a fitness landscape. The fitness landscape is defined by a fitness function that is typically composed of an objective function and an appended penalty function if the problem is constrained. Unlike calculus-based methods, GAs are developed as a framework for a global search of the design space [7].

GA were formulated by Holland in 1975 as a computational technique to artificially model biological evolution [8]. GA as search and optimization routines were popularized by Goldberg’s 1989 publication GA in Search, Optimization, and Machine Learning [9]. The GA and its variations, along with similarly inspired genetic programming, evolutionary programming, and evolution strategies, constitute the family of evolutionary algorithms (EA) [10]. Evolutionary algorithms are then grouped into the larger category of biologically-motivated evolutionary computation, which includes techniques such as ant colony optimization, particle swarm optimization, and differential evolution.

“Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime.” (- Salvatore Mangano Computer Design, May 1995).

Provide efficient, effective techniques for optimization and machine learning applications
♦ To understand the adaptive processes of natural systems
♦ To design artificial systems software that retains the robustness of natural systems

I.2 Components of a GA

A problem to solve, and...
• Encoding technique (gene, chromosome)
• Initialization procedure (creation)
• Evaluation function (environment)
• Selection of parents (reproduction)
• Genetic operators (mutation, recombination)
• Parameter settings (practice and art)

This way of search technique is shown in figure 1
A GA for a particular problem must have the following five components as shown in figure 2:

- a genetic representation for potential solutions to problem
- a way to create an initial population of potential solutions
- an evaluation function that plays the role of the environment, rating the solutions in terms of their fitness values
- genetic operators that alter the composition of children
- Values for various parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc.)

2.3 The GA Cycle of Reproduction

Is the biological process by which new "offspring" individual organisms are produced from their "parents". Specify how the genetic algorithm creates children for the next generation as shown in figure 3. The steps in described as follow

- Initialize population with random chromosomes
- Decode chromosome into phenotype
- Evaluate fitness
- Select new generation probabilistically based on fitness
- Crossover chromosomes of parents
- Mutate chromosomes

There is significant benefit for applying the genetic algorithm as the optimization technique. The genetic algorithm is an optimization algorithm which has both global search and local search abilities. With the crossover operation, we implement the local search. With mutation operation, we implement the global search, which checks randomized candidate other than similar candidates which we produce with crossover. In conventional optimization algorithm, most algorithms have only one search method, not both. Also the conventional optimization technique uses the sequential evaluation, in which the algorithm generates only single candidate, evaluates and compares with the current system. Example of sequential optimization is simulated annealing [13] [14]. For the genetic algorithm, we use the parallel optimization, which generates multiple candidates, evaluate, and compares with the previous population. We can find better candidates more efficiently since we check more candidates simultaneously and choose better candidates.

II. CHEMICAL REACTOR

Stirred Tank Reactor (CSTR) is an important component in chemical process and offering a diverse range of researches in the area of the chemical and control engineering. Various control approaches have been applied on CSTR to control its dynamics. The problem of controlling of CSTR is considered as an attractive and controversial issue, especially for control
engineers, corresponding to its nonlinear dynamic. Most of the
conventional controllers are restricted just for linear time
invariant system applications [15]. However, in real
environment, the nonlinear characteristics of the systems and
their functional parameters changes, due to wear and tear,
cannot be neglected. Furthermore, dealing the systems with
uncertainties in real applications, is another subject which
must be noticed. In this way, the role of the adaptive and
intelligent controllers, by the capability of the overcoming the
aforementioned points are of importance. A chemical reactor
is adopted that finds typical applications in chemical industry.
The chemical reactor is a very rich example of MIMO systems
with complex nonlinear behaviour and sensitivity to
parameters uncertainty.

II.1 statement of the problem
It is required to control the temperature of the reactor via
adjusting the position of the control valve.

II.2 Mathematical Model of a Chemical Reactor
Continues Stirred Tank Reactor (CSTR) as shown in figure
4. It is a jacketed-type reactor, and it's assumed that:

II.2.1 CSTR Model
The system has three states given by:
\[ X = [x_1 \ x_2 \ x_3]^T = [C_A \ T \ T_c]^T \]
\[ Y = [0 \ 1 \ 0] X \]
Where:
- \( C_A \) is the concentration of the reactant in the reactor
- \( T \) is the temperature of the reactor
- \( T_c \) is the jacket temperature
- \( Y \) is the output of the model

A Cyclopentadine (Input flow concentration of A)
B Cyclopetenol (output flow concentration of B)

A flow stream A is fed to the reactor. A catalyst is placed inside the reactor. The liquid inside the reactor is perfectly mixed and sent out
through the exit valve. The jacket surrounding the reactor also has feed and exit streams. The jacket is assumed to be perfectly adjacent with the tank and
at a lower temperature than the reactor [16], [17]. The mathematical model equations are obtained by
a component mass balance (1), energy balance
principle (2) in the reactor and energy balance
principle (3) on the jacket.

- Balance of mass of reactant A:
  \[ \frac{dC_A}{dt} = \frac{F}{V} (C_{A_i} - C_A) - K_o e^{-\frac{\Delta H}{R(T + \Delta T)}} C_A^2 \]
- Energy balance on reactor contents:
  \[ \frac{dT}{dt} = \frac{F}{V} (T_i - T) - \frac{\Delta H}{\rho c_p} K_o e^{-\frac{\Delta H}{R(T + \Delta T)}} C_A^2 - \frac{UA}{\rho c_p} (T - T_c) \]
- Energy balance on jacket:
  \[ \frac{dT_c}{dt} = -\frac{UA}{\rho c_p c_{f_c}} (T - T_c) + \frac{F_c}{V_c} (T_c - T_{ci}) \]

Where:
- \( C_A \) Concentration of the reactant in the reactor
- \( T \) Temperature in the reactor
- \( T_c \) Jacket temperature
- \( C_{A_i} \) Concentration of the reactant in the feed
- \( T_i \) Temperature in the feed
- \( T_{ci} \) Coolant inlet temperature
- \( F \) Feed rate
- \( V \) Reactor volume
- \( K_o \) Arrhenius frequency parameter
- \( \Delta H \) Heat reaction (assumed constant)
- \( \rho \) Density of reactor content
- \( C_p \) Heat capacity of the reactants
- \( U \) Overall heat transfer coefficient
- \( A \) Heat transfer area
- \( V_c \) Jacket volume
- \( \rho_c \) Density of the coolant
- \( C_{pc} \) Heat capacity of the coolant
- \( F_c \) Coolant rate
- \( F_{cmax} \) Maximum flow through the control valve
Thus the system is seen to be third order and heavily nonlinear as shown in figure 6. The idea is to regulate the output to set point temperature using only one control signal, which is the signal applied to the control valve.

![Figure 6: system description](image)

A complete description of the parameters involved in deriving the mathematical model of the reactor, their units, as well as their nominal values is given in table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_A$</td>
<td>Kg/mole/m^3</td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>$T_c$</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>$C_{Ai}$</td>
<td>Kg/mole/m^3</td>
<td>2.88</td>
</tr>
<tr>
<td>$T_{ci}$</td>
<td>°C</td>
<td>27</td>
</tr>
<tr>
<td>$F$</td>
<td>m^3/s</td>
<td>7.5E-3</td>
</tr>
<tr>
<td>$V$</td>
<td>m^3</td>
<td>7.08</td>
</tr>
<tr>
<td>$K_o$</td>
<td>m^3/s-kg/mole</td>
<td>7.44E-2</td>
</tr>
<tr>
<td>$\Delta H_R$</td>
<td>J/kg/mole</td>
<td>-9.86E7</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Kg/m^3</td>
<td>19.2</td>
</tr>
<tr>
<td>$C_p$</td>
<td>J/kg/mole-°C</td>
<td>1.815E5</td>
</tr>
<tr>
<td>$U$</td>
<td>J/s-m^2-°C</td>
<td>3.55E3</td>
</tr>
<tr>
<td>$A$</td>
<td>m^2</td>
<td>5.4</td>
</tr>
<tr>
<td>$V_c$</td>
<td>m^3</td>
<td>1.82</td>
</tr>
<tr>
<td>$p_c$</td>
<td>Kg/m^3</td>
<td>1000</td>
</tr>
<tr>
<td>$C_{inc}$</td>
<td>J/kg/mole-°C</td>
<td>4.184E3</td>
</tr>
<tr>
<td>$F_c$</td>
<td>m/s</td>
<td></td>
</tr>
<tr>
<td>$F_{c_{max}}$</td>
<td>m/s</td>
<td>0.02</td>
</tr>
<tr>
<td>$A$</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>$E$</td>
<td>J/kg/mole</td>
<td>1.182E7</td>
</tr>
<tr>
<td>$R$</td>
<td>J/kg/mole-°K</td>
<td>8314.39</td>
</tr>
</tbody>
</table>

Table 1: system parameters

II.3 Using noise at low temperature

In this case a low level of temperature used as an input to the chemical reactor as shown in figure 7.

![Figure 7: input signal with noise](image)

As shown the noise is random and ripple ratio about 40% from the input value.

II.3.1 Using proportional controller

In the proportional control algorithm, the controller output is proportional to the error signal, which is the difference between the set point and the process variable. Figure 6 show the output after using P controller. (K_p=20.8718473357658)

II.3.2 using proportional-integral controller

A PI Controller (proportional-integral controller) is a special case of the PID controller in which the derivative (D) of the error is not used. The result of using this type of controller is shown in figure 9.

![Figure 8: system with P controller](image)

![Figure 9: system with PI controller](image)
Where \((K_p=20.87, K_i=-9.8317)\)

**II.3.3 using PID controller**

Proportional-integral-derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial. A PID controller calculates an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs. The result of using this type of controller is shown in figure 10 as shown blow. Where \((K_p=20.87, K_i=-9.8317, K_d=0.183)\)

![Figure 10: system with PID controller.](image)

As a result \(t_s=192.21\) sec, and \(t_r=1.31\) sec

**II.3.4 using Genetic Algorithm**

To describe the GA optimization process, consider the system as shown if figure 11. At the beginning of the process, the initial populations comprise a set of chromosomes that are scattered all over the search space. The initial population may be randomly generated. However, in all experiments, the population consists of 25 chromosomes which are all randomized initially. Thus, the use of heuristic knowledge of the controller is minimized.

![Figure 11: PID controller design using GA](image)

**II.3.4.1 Parameters of GA**

GA has many parameters like, population size, probability of crossover, probability of mutation, the way you encode your variable, etc. as shown in figure 12

![Figure 12: GA steps flowchart](image)

**II.3.4.1.1 Population size**

Specifies how many individuals there are in each generation. The population consists of 25 chromosomes, each one represent a temperature.

**II.3.4.1.2 Selection**

Specify how the genetic algorithm chooses parents for the next generation, here the selection is Roulette Wheel. The size of the section in the roulette wheel is proportional to the value of the fitness function of every chromosome - the bigger the value is, the larger the section is.

**II.3.4.1.3 Crossover**

Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability. The type of crossover used here is discrete recombination.

**II.3.4.1.4 Mutation**

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can
result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. Mutation is an important part of the genetic search as help helps to prevent the population from stagnating at any local optima. The type of mutation that used here is Non Uniform mutation

II.3.4.1.5 stopping criteria
Stall generations: The algorithm computes the specified number of generations with no improvement in the fitness function. Here no of generations was 150.

II.3.4.1.6 Range of parameters
Range that bounded the generation of the PID parameters (Kp,Ki,Kd). here the range was from -20 to 40
Applying the GA to the system this will led to the PID parameters (Kp,Ki,Kd). And the transfer function of the PID controller.

The system is shown in figure 13 after applying the Genetic Algorithm technique to compute PID parameters

II.4 Using noise at high temperature
In this case a high level of temperature used as an input to the chemical reactor as shown in figure 14.

II.4.1 Using proportional controller
The result of using this type of controller is shown in figure15.

II.4.2 using PI controller
The result of using this type of controller is shown in figure16.

II.4.3 using PID controller
The result of using this type of controller is shown in figure 17 as shown blow
II.4.4 using Genetic Algorithm

The results of using genetic algorithm to control the parameter of PID is shown in figure 18 as shown blow

![Figure 18: system with GA](image)

II.4 Summarized Results

As a results of the above figures, the results could be compared in table 2. That shown the noise percentage of input and the ripple ratio by using different types of controllers.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Noise 40%</th>
<th>Noise 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>10.2%</td>
<td>4.34%</td>
</tr>
<tr>
<td>PI</td>
<td>9.74%</td>
<td>3.65%</td>
</tr>
<tr>
<td>PID</td>
<td>7.36%</td>
<td>2.98%</td>
</tr>
<tr>
<td>GA</td>
<td>1.24%</td>
<td>1.25%</td>
</tr>
</tbody>
</table>

Table 2: comparative results

Table 3 summarize the values of PID parameters (Kp, Kv, Ka)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Kp</th>
<th>Kv</th>
<th>Ka</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>20.87185</td>
<td>0.183389</td>
<td>-9.8317</td>
</tr>
</tbody>
</table>

Table 3: PID values

III. CONCLUSION

The proposed GA is tested by using Math lab Simulink program and its performance is compared to a different temperature & concentration. The paper demonstrated that while the GA controller exhibits superior control in the presence of nonlinearities.

This paper illustrates the control of non-linear system CSTR (Continuous Stirred Tank Reactor). And the results prove that GA controllers are appropriate under non-linear difficulties.

REFERENCES