Study on Self-Adaptive Control System - View of Internal Adaptation

Hussam Mohammed-ElHassan Ibrahim Dafaalla[1], Dalia Mahmoud[2].

[1]Alneelain University, Faculty of Engineering, Control Systems Engineering, Khartoum, Sudan

[2]Alneelain University, Faculty of Engineering, Assistant Professor, Head of Control Systems Department, Khartoum, Sudan

Abstract: *In world's manufacturing industry rapid development, using of the control system has become vital issue and more in demand for many reasons such like to keep up with market rising demand of goods and competition of its' quality. However, with the complicating of the production system, the traditional control system has not been able to meet the needs of the current industry. Bringing the genetic algorithm of the neural network into the control system can effectively solve this problem. In this paper we discuss the neural network, genetic algorithm principle, operation procedures and the characteristics; then, analyzes the principle and lack of conventional PID controller; finally, effectively combines genetic algorithm and controller together, forming a closed loop, strengthening the control of parameters, and giving a code description of the genetic algorithm. This paper plays a certain positive role for industrial engineers and programmers.*

Keywords: Neural Network; Genetic Algorithm; Self-Adaptive Control

1. Introduction

Although the characteristics of linear coefficients of any controlled object can be known with its' process but, the actual operation, the controlled object and its process are difficult to pre-determined in advance.

There are certain conditions where a mathematical model would be unclear such like.

- The environment changes; when the conditions not suitable then, the related model coefficients and parameters will cause a series of changes. The conventional regulator under this variation would not be able to meet the changes brought by the system. And here where the adaptive control system comes in need to protect its control quality.

2. Genetic Algorithm

A genetic algorithm is a type of bionic algorithm in the macro sense. It would simulate the Darwin's law of -survival of the fittest, and the genetic variation theory of Mendel by the computer.

Through an iterative method it will generate an exploratory algorithm (testing results).

Coding technique performed by genetic algorithm changes the genetic problem to be solved into a large number of recognizable chromosomes for computer. This kind of algorithm mainly carries out the mode recombination by crossover operators, the mode judgment by selection operator. The rule of survival of the fittest will be adopted, and the chromosomes would there for continuously be enhanced to reach the optimal result, to be more rational and equal, to achieve the purpose of adaptation operation.

3. Neural Network

What is a neural network? The simplest definition of a neural network, more properly referred to as an 'artificial' neural network (ANN), is provided by the inventor of one of the first neurocomputers, Dr. Robert Hecht-Nielsen. He defines a neural network as: "...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.

4. Operation procedures of Genetic Algorithm

The elements that the operation mainly involves are: generation and setting of the initial population, fitness function, parameters encoding, genetic operation (ex. selecting, crossing, mutating ...etc), setting of termination conditions and setting of a threshold. The genetic algorithm is a framework that is not targeted at any specific problems. It can be applied in many fields and has strong robustness. The operation procedure is shown in Figure 1:
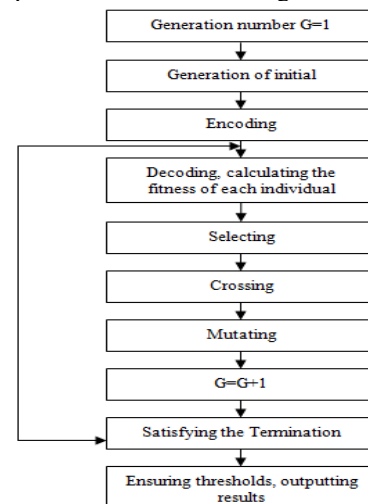

Figure 1: Procedure of Genetic Algorithm

The description of the above main steps is:
(1) Initializing population. First, it determines the size of the population scale. In the search number-group space of the whole problem, it randomly generates the initial population to be searched, and the genetic generation number is set to 1.
(2) Determining code. The group is converted from the space waiting to be solved to encoding space being able to be identified by the algorithm.
(3) Designing fitness function. This is a merit-based criterion, to protect the better performed individuals continuously passing down by judging the individuals.
(4) Using the operating of selecting, crossing and mutating to design genetic operators, it constantly gets new populations.
(5) The operation selects the individual meeting the termination conditions, to determine its threshold, and output the result.

4.1 Characteristic of Genetic Algorithm

The genetic algorithm is the earliest produced, most influential and most widely used evolutionary algorithm. Its main features are mainly manifested in the following aspects:
(1) The processing of chromosome space is to encode the parameters of the individual, but not the specific individual;
(2) High searching efficiency, high parallelism to group operating;
(3) With strong adaptability and scalability, lower requirements for fitness function;
(4) Better ability of globally solving optimum, maximum convergence for the noise and non-continuous problems;
(5) Simple in thinking and easy to implement.

5. PID Controller

The simplified structure of PID controller would help in the control operation and it would make it easy to implement. That's why the use of PID is widely spread in chemical industries, machinery, metallurgy, and other areas,

5.1 How PID operates?

- Analyzing the operating object to obtain the relevant mathematical model to determine the PID parameters.
- Optimizing the PID parameters by relevant mathematical analysis.

5.2 The main deficiency in that:

- It can not achieve the response of parameters within a closed-loop system.
- It can't satisfy the nonlinear processing.
- Lagging problems on dealing complex questions, time-varying uncertainties, etc.

5.3 PID controller Principle

What we are seeing in industries nowadays is the PID controllers linear type. The PID algorithm is mainly completed by operation of proportion, integral and derivative. First, the data of set point is input to the controller. The error generated is analyzed by the PID algorithm. The results will be sent into the process (controlled object). Comparing the generated results with the ideal state, the individuals and data information which interfere the controller will be processed and analyzed at the next time under its control. This will be looping till generating an optimal result. The specific principle is shown in Figure 2:
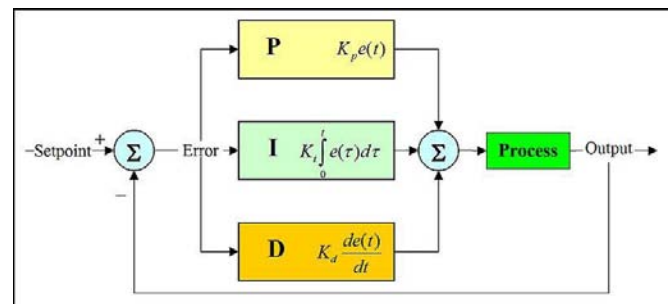

Figure 2: PID controller Schematic diagram

Throughout the correction of control algorithm, the main role proportion part *(P)* plays is to reduce the deviation, the main role integral part of ({) plays is to eliminate the static error, and the main role derivative part *(d /dt)* plays is to reduce system adjusting time by leading in the early correction signal.

6. Deficiencies of Conventional PID Controller System

As we discussed before the main three processes which PID algorithm depends on are (proportion, integral, and derivative). In many cases when the proportional coefficient is too big, it will cause system instability; and when the integral coefficient is too big or too small, this will cause the system dynamic performance deterioration and anti-interfere ability weakened. In today's world with the increasing complexity of industrial control, the conventional PID controllers are more and more challenged on the correlation coefficient selection. In many cases, it is difficult to achieve the desired control effect of the system by the controller.

7. Self Adaptive Control based on Genetic Algorithm, how it works?

To initially setup the parameters of the controlled object, it will start using the parameter identification mechanism, and then it will feed the mathematical object model with measures which is most original and with some identification ability. Setting the parameters of a controlled object according to the parameters recorded in the internal system, it will get the first group of PID parameters of the system. This is to ensure that the system can properly run at the first time. Over time, the performance indexes of the system are constantly transforming. This is where the PID parameters of the system need to be adjusted. In normal operating conditions we have two choices after the parameter identification mechanism analyzes the current parameters and

system operating status.

- If satisfied, it will continue to implement.
- If not satisfied, it will reset the system parameters. This set value will primarily be determined to its' initial value of PID controller. Every solved result is directly related to the previous operating parameter. The operation will get a group of optimal solutions or sub-optimal solutions by genetic algorithm, to run the relevant program in accordance with the related search algorithm. There is no system that will not stop, for the great waste of time and space resources. Every interval of time, the value is determined according to the actual needs of production environment. When the calculation is not required, it will close the switch K2, and open the switch Kl. Thus, in the condition of absence of initial solution, the system no longer repeats the genetic algorithm. When it needs to adjust, it will close the K1 and open the K2. The system has an adaptive capacity, while it will adjust the system parameters according to the actual needs of the situation to ensure that the system has always been in a reliable and secure operating status. The specific principle is shown in Figure 3:
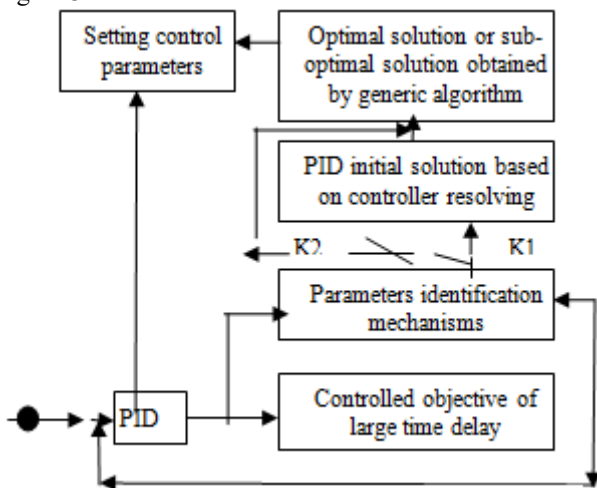

Figure 3: Block diagram of the genetic algorithm of the adaptive controller

For the controller parameter optimization, it will convert it to a constrained optimization problem, where the target parameter is:

$$J = \int_o^x t\,|\,e\,|\,dt \qquad (1)$$

The three parameters of the controller are K_p, K_i and K_d, where K_{p0}, K_{i0} and K_{d0} are the initial parameters, and α is a coefficient of the search range. The range is as follows:

$$\begin{cases} (1-\alpha)K_{p0} < K_p < (1+\alpha)K_{p0} \\ (1-\alpha)K_{i0} < K_i < (1+\alpha)K_{i0} \\ (1-\alpha)K_{d0} < K_d < (1+\alpha)K_{d0} \end{cases} \qquad (2)$$

PID Algorithm code example 1:

```
/*working variables*/
unsigned long lastTime;
double Input, Output, Setpoint;
double errSum, lastErr;
double kp, ki, kd;
void Compute()
{
```

```
   /*How long since we last calculated*/
   unsigned long now = millis();
   double timeChange = (double)(now - lastTime);

   /*Compute all the working error variables*/
   double error = Setpoint - Input;
   errSum += (error * timeChange);
   double dErr = (error - lastErr) / timeChange;

   /*Compute PID Output*/
   Output = kp * error + ki * errSum + kd * dErr;

   /*Remember some variables for next time*/
   lastErr = error;
   lastTime = now;
}

void SetTunings(double Kp, double Ki, double Kd)
{
   kp = Kp;
   ki = Ki;
   kd = Kd;
}
```

PID Algorithm code example 2:

```
#ifndef PID_REG_H
#define PID_REG_H
#ifdef __cplusplus
extern "C" {
#endif
// These defines are needed for loop timing and PID
controller timing
   #define TWENTY_SECONDS (400)
   #define TEN_SECONDS (200)
   #define FIVE_SECONDS (100)
   #define ONE_SECOND (20)
   #define T_50MSEC (50) // Period time of TTimer in
msec.
   #define GMA_HLIM (100.0) // PID controller upper limit
[%]
   #define GMA_LLIM (0.0) // PID controller lower limit
[%]
   typedef struct _pid_params
   {
   double kc; // Controller gain from Dialog Box
   double ti; // Time-constant for I action from Dialog Box
   double td; // Time-constant for D action from Dialog Box
   double ts; // Sample time [sec.] from Dialog Box
   double k_lpf; // Time constant [sec.] for LPF filter
   double k0; // k0 value for PID controller
   double k1; // k1 value for PID controller
   double k2; // k2 value for PID controller
   double k3; // k3 value for PID controller
   double lpf1; // value for LPF filter
   double lpf2; // value for LPF filter
   int ts_ticks; // ticks for timer
   int pid_model; // PID Controller type [0..3]
   double pp; // debug
   double pi; // debug
   double pd; // debug
   } pid_params; // struct pid_params
```

```
//--------------------
// Function Prototypes
//--------------------
void init_pid2(pid_params *p);
void pid_reg2(double xk, double *yk, double tset,
pid_params *p, int vrg);
void init_pid3(pid_params *p);
void pid_reg3(double xk, double *yk, double tset,
pid_params *p, int vrg);
void init_pid4(pid_params *p);
void pid_reg4(double xk, double *yk, double tset,
pid_params *p, int vrg);
#ifdef __cplusplus
};
#endif
#endif
```

8. Conclusion

This paper researches the genetic algorithm of neural network usage in the controller. The automatic control uses in the feed-forward neural network. Self-adaptive control can combine with machine learning, artificial intelligence and other algorithms. Due to space limits, this paper doesn't give the more examples of the applications of genetic algorithms and controller. With hope that readers may have ability to grasp the idea on this basis and can combine with actual to develop the adaptive controller fitted the enterprise itself.

References

[1] Wang Xiaozhe, Gu Shusheng. The research of multivariable decoupling control method based on neural network, Control and Decision (1999).
[2] Wang Zhengzhi, Bo Tao, Evolution algorithms. Changsha: National University of Defense Techonology Press (2000).
[3] Zou wei,Wei ming,Zhang zhu-ping,The Intelligent and Self-adaptive control technology Research of Power Amplifier in Communication Counter Measure, Journal of CAEIT, Vol.7, No.3 (2012).
[4] Li Qingquan. Adaptive control system theory design and application. Beijing: Science Press (1990).

Author Profile

Hussam Mohammed-ElHassan Ibrahim Dafaalla received his B.Sc. Engineering from European University of Lefke, Turkish Higher Education Board YÖK. He has over 5 years experience in Telecommunication industry at Saudi Arabia. Now he is doing his supplementary research on Master Degree at Sudan.

Dalia Mahmoud Assistant Professor and currently Head of Control Systems Department at Alneelain University –Faculty of Engineering in Sudan. Dalia holds a PhD degree and she has more than 10 years Experience in higher Education field.