We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



3

Enhancing Control Systems Response Using Genetic PID Controllers

Osama Y. Mahmood Al-Rawi
Electrical and Electronic Engineering Department, Gulf University, Kingdom of Bahrain

1. Introduction

The acronym PID stands for Proportion-Integral-Differential control. Each of these, the P, the I and the D are terms in a control algorithm, and each has a special purpose. Conventional PID controller has a simple structure and can provide satisfactory performance for many control problems. For process control systems, in particular, more than 95% of the controllers used are of PID type (Ogata, 2002). This is because it is easily applicable to field use, and also easily adjustable. The process of stabilization or improving the performance of the system is achieved by attaching a controller or compensator to the closed loop system. There are different types and structures of controllers. However, the controller cannot deal fully with nonlinear and/or time varying systems. In the design of a PID controller, the main difficulty has been how the control gains should be selected for good control performance, especially where the process to be controlled is of high order with complex nonlinear dynamics. This problem is made more difficult by the need to keep these gains tuned, during operations, to ensure robust performance in the face of plant parameter variations, load or environment disturbances and measurement noise. Genetic algorithms (GAs) are one of the efficient tools that are employed in solving optimization problems (Rothlauf, 2006). Because of the simplicity and robustness, PID controllers are frequently the used controllers in industries (Almeida et al., 2005). Parameters adjustment of PID controllers is an old challenge in the field of control system design. Some of methods have been proposed to select the PID coefficients, but they are not completely systematic methods and result in a poorly tuned controller that needs some trail and error. So far, finding new methods to automatically select PID parameters was interest of researches (Ho, 2003), (Almeida et al., 2005), (Nasri et al., 2007), and (Mansour, 2011). However in this chapter, the selection of optimal PID controller coefficients are introduced and applied based on genetic algorithms technique. Therefore it is possible to find genetically the optimal controller coefficients, K_p, K_i, and K_d with the constraint of minimizing absolute or square error signal to achieve system stability and to enhance system performance. However in feedback control systems the most important requirement is the stability of the whole system (Bartoszewicz, 2011). Another requirement is the speed, i.e. in the case of a changing reference value or a disturbance; the control error should be eliminated as soon as possible. Finally, simulation results will be implemented using MATLAB/SIMULINK to test system performance and stability. Due to the fact that in the last two decades MATLAB became the standard computing environment for engineers from different fields of science, a

number of researchers addressed the issue of control system toolbox in MATLAB/SIMULINK.

2. Feedback control systems

Feedback is a powerful idea, which is used extensively in natural and technical systems. The principle of feedback is very simple; base correcting actions on the difference between desired and actual performance. In engineering feedback has been rediscovered and patented many times in many different contexts (Rodić, 2009) and (Bartoszewicz, 2011). Different types of controllers are used to improve the output of feedback control system. Figure 1 shows one of the main types of cascade or series controller, which are used in many different control system applications. In general there are three major types of continuous-time controllers (Ogata, 2002):

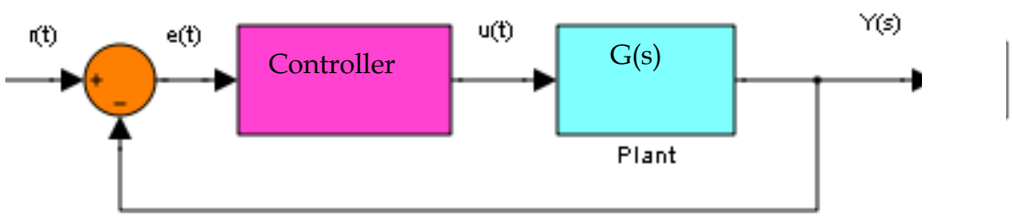


Fig. 1. Block diagram of feedback control system with cascade controller.

2.1 Proportional control action

This is the simplest type of continuous control law. The controller output $u(t)$ is made proportional to the actuating error signal $e(t)$. The proportionality constant is called the proportional gain, K_p, and the control law can be written as:

$$u(t) = K_p e(t) \tag{1}$$

where $e(t) = r(t) - y(t)$ for unity feedback system, $e(t)$ is the error signal, $r(t)$ is the reference input of the system, and $y(t)$ is the output of the system. Taking Laplace-transform of equation (1) yield,

$$\frac{U(s)}{E(s)} = K_p \tag{2}$$

One important thing to be noted about proportional control is that is incapable of maintaining the output steady state value at the desired value. This is clear from the equation above, note that as long as a non-zero actuation is required to maintain the system at the desired value, the error cannot be zero. Mathematically, $u(t)= 0$, therefore $e(t) = u(t)/K_p = 0$. As the value of the gain increased, the steady state error will be decreased. However, K_p is limited by the dynamics of the system. Therefore the value of K_p will have to be arrived at by compromising between the steady state error and the dynamic stability of the system.

2.2 Integral control action

In a controller with integral control action, the value of the controller output is changed at a rate proportional to the actuating error signal. If zero steady state error is desirable, this means a control mode that is a function of the history of error will be accumulation. The longer the error persists, the stronger the control action should be to cancel it. The mathematical operation of integration is a means of implementing this action as:

$$\frac{du(t)}{dt} = K_i e(t) \tag{3}$$

or

$$u(t) = K_i \int_0^t e(t)dt \tag{4}$$

where K_i is an integral gain constant. In Laplace-transformed quantities, the transfer function of the integral controller is

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \tag{5}$$

It can be seen that the integral action is capable of reducing the steady state to zero. This is because even though the steady state error is reduced to zero, the integral controller is still capable of maintaining some actuation (i.e $u(t)$ = 0) because of the past history of error values. Note that integral control is usually combined with proportional control to give a PI controller, and the control law will be written as:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt \tag{6}$$

where $K_i = \frac{K_p}{T_i}$, and T_i is the integral time. In Laplace-transformed quantities, the transfer function of the PI controller is

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \tag{7}$$

The PI controller has two tuning parameters, namely K_p and K_i. It should be noted that the integral action has a de-stabilizing influence on the system and therefore it may be necessary to reduce the proportional gain somewhat if the integral action causes too much oscillation (Ogata, 2002).

2.3 Derivative control action

The use of integral action is sufficient to reduce the steady state error to zero. However, the dynamic or the transient response may still be poor because of large oscillations, overshoots

etc. The derivative control action, sometimes called rate control, is where the magnitude of the controller output is proportional to the rate of change of the actuating error signal. Derivative control can be used in such cases to stabilize the dynamic behavior of the system. Mathematically the derivative control law can be written as

$$u(t) = K_d \frac{de(t)}{dt} \tag{8}$$

where K_d is the derivative gain constant. Taking Laplace-transform of equation (8) yield,

$$\frac{U(s)}{E(s)} = K_d s \tag{9}$$

Thus the derivative action can be used to create damping in a dynamic system and thus stabilize its behavior. It must however be noted that derivative action slows down the initial response to the system. While derivative control action has the advantage of being anticipatory, it has the disadvantages that it amplifies noise signals and may cause a saturation effect in the actuator. However the derivative control action can never be used alone because this control action is effective only during transient periods. However derivative control is usually combined with proportional control to give a PD controller (Ogata, 2002), and the control law can be written as:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \tag{10}$$

Substituting for $K_d = K_p T_d$ and taking Laplace-transform of equation (10) yields,

$$\frac{U(s)}{E(s)} = K_p(1 + T_d s) \tag{11}$$

where T_d is the derivative time. In addition the combination of proportional control action, integral control action, and derivative control action is termed as Proportional-plus-Integral-plus-Derivative control action. This combined action has the advantages of each of the three individual control actions. The equation of a controller with this combined action is given by

$$u(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)dt + K_p T_d \frac{de(t)}{dt} \tag{12}$$

Taking Laplace-transform of equation (12), yield the transfer function of PID controller

$$\frac{U(s)}{E(s)} = K_p(1 + \frac{1}{T_i s} + T_d s) \tag{13}$$

or

$$\frac{U(s)}{E(s)} = K_p + \frac{Ki}{s} + K_d s \tag{14}$$

3. Design and tuning rules of PID controllers

In general, the Design Procedure for determining parameters of the PID controller is a trail and error approach. After obtaining a mathematical model of the control system and adjust the parameters of a compensator to improve system performance. The most time consuming part of such work is the checking of the system performance by analysis with each adjustment of the parameters. The designer should make use of a digital computer to avoid much of the numerical drudgery necessary for this checking (Ogata, 2002). Once a satisfactory mathematical model has been obtained, the designer must construct a prototype and test the open loop system. If the absolute stability of the closed loop is assured, the designer closes the loop and tests the performance of the resulting closed loop system. Because of the neglected loading effects among the components, nonlinearities, distributed parameters, and so on, which were not taken into consideration in the original design work (Mansour, 2011), the actual performance of the prototype system will probably differ from the theoretical predictions (Rodić, 2009). Thus the first design may not satisfy all the requirements on performance. By trail and error, the designer must make changes in the prototype until the system meets the desired specifications. However the process of selecting the controller parameters to meet given performance specifications is known as controller tuning. The controller tuning greatly affects the control system properties, such as robustness to disturbances and noise, performance and robustness to delays (Mansour, 2011). For example, figure 2 shows a block diagram of a simple cascade PID control of a plant. If a mathematical model of a plant can be derived, then it is possible to apply various design techniques for determining parameters of the controller that will meet the transient and steady-state specifications of the closed-loop system. However, if the plant is so complicated that its mathematical model cannot be easily obtained, then analytical approach to the design of PID controller is not possible (Ogata, 2002). Then we must resort to experimental approaches to tuning of PID controllers.

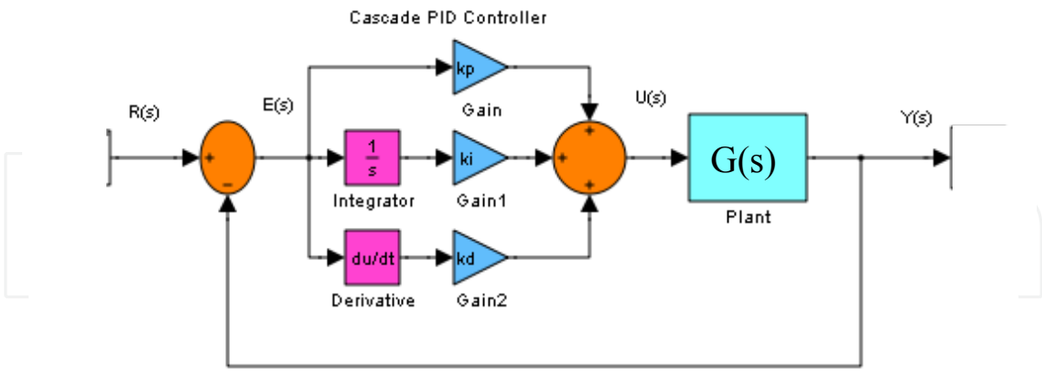


Fig. 2. Closed-loop control system with cascade PID controller.

Therefore, an alternative approaches in designing control systems by using graphical methods such as root-locus, Bode plot or by using frequency response methods. It is important to note that using such graphical methods the final results is not unique, because the best or optimal solution may not be precisely defined if the time-domain specifications or frequency-domain specifications are given. Thereafter, many different rules are suggested for tuning PID controllers, such as Ziegler and Nichols rules. However Ziegler and Nichols

suggested rules when mathematical models of plants are not known (Ogata, 2002). In addition these rules can, of course, be applied to the design of systems with known mathematical models.

3.1 Controller performance evaluations

In order to evaluate the performance of a closed-loop control system, a cost criterion must be set. The most common ones are in PID controller design methods, the most common performance criteria are J_{IAE} (Integral of Absolute Error), J_{ISE} (Integral of Square Error), J_{ITAE} (Integral of Time-Weighted Absolute Error), and J_{ITSE} (Integral of Time-Weighted Square Error). They are given, respectively, in (15) - (18)

$$J_{IAE} = \int_0^\infty |e(t)|\,dt \tag{15}$$

$$J_{ISE} = \int_0^\infty (e(t))^2\,dt \tag{16}$$

$$J_{ITAE} = \int_0^\infty t|e(t)|\,dt \tag{17}$$

$$J_{ITSE} = \int_0^\infty t(e(t))^2\,dt \tag{18}$$

These four integral performance criteria in the frequency domain have their own advantage and disadvantages. For example, disadvantage of the J_{IAE} and J_{ISE} criteria is that its minimization can result in a response with relatively small overshoot but a long settling time because the J_{ISE} performance criterion weights all errors equally independent of time. Furthermore by using the J_{ITSE} performance criterion this tends to overcome the disadvantages of the J_{ISE} criterion.

4. Genetic algorithms

4.1 Introduction

In the early 1960s Rechenburg (1965) conducted studies at the technical university of Berlin in the use of an evolutionary strategy to minimize drag on a steel plate (Goldberg, 2002). Genetic algorithms were used by Holland (1975) and his students at the University of Michigan in the late 1970s and early 1980s to analyse a range of engineering problems (Goldberg, 2002) and (Franz, 2006). In particular, Goldberg (1983) used genetic algorithms to optimize the design of gas pipeline systems. Genetic algorithms are one of the efficient tools that are employed in solving optimization problems (Rothlauf, 2006). The basic idea of genetic algorithm is as follow (Goldberg, 2002), the genetic pool of a given population potentially contains many solutions, or candidate solutions. These solutions are not active because the genetic combination on which it relies is split between several subjects. Only the

association of different genomes can lead to the solution. Optimization in genetic algorithm is based on optimization of a fitness function which is a function of environment individuals or genes (Rothlauf, 2006). Each new generation is generated by applying reproduction, crossover and mutation operand on old generation. Then in new generation good genes that lead to better fitness function have more chance to survive. Finally after some generations the optimal solution will be attained.

4.2 Genetic operators

In such algorithms, the fittest among a group of genetic algorithms creatures can survive and constitute a new generation (Rothlauf, 2006). In every new generation, a new offspring is created using features of the fittest individuals of the current population. A fitness value is assigned to each solution representing the ability of an individual to 'compete'. By combining information from the chromosomes, selective breeding of individuals is utilized to produce offspring better than the parents. The goal is to produce an individual with the fitness value close to the optimal. The basic genetic operators are (Goldberg, 2002):

4.2.1 Selection

The selection mechanism favors the individuals with high fitness values. It allows these individuals better chance for reproduction into the next generation while reducing the reproduction ability of least fitted members of population. Fitness of an individual is usually determined by an objective function. However, there are many different types of selection methods, such as Roulette wheel and tournament selection which are widely used in different genetic algorithms applications (Goldberg, 2002).

4.2.2 Crossover

The crossover operator divides a population into the pairs of individuals and performs recombination of their genes with a certain probability. If one-point crossover is performed, one position in the individual genetic code is chosen (Goldberg, 2002). All gene entries after that position are exchanged among individuals. The newly formed offspring created from this mating are put into the next generation. Recombination can be done at many points, so that multiple portions of good individuals are recombined, this process is likely to create even better individuals.

4.2.3 Mutation

When using mutation operator a portion of the new individuals will have some of their bits flipped with a predefined probability. The purpose of mutation is to maintain diversity within the population and prevent premature convergence. The usage of this operator allows the Individuals in genetic algorithms are usually in the form of character strings that are analogous to the chromosome found in DNA (Rothlauf, 2006). Each individual represents a possible solution within a search space. A number of individuals constitute a population. The individuals in the population are then made to go through a process of evolution, in order to produce a new generation of individuals that is closer to the optimal solution (Goldberg, 2002). The process of evolution is based on the following principles:

- Individuals in a population compete for resources and mates.
- The most successful individuals in each generation will have a chance to produce more offspring than those individuals that perform poorly.
- Genes from good individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent. Thus each successive generation will become more suited to their environment.

4.3 Genetic algorithms applications to control system design

Because of the simplicity and robustness, PID controllers are frequently used controllers in industries (Mansour, 2011). Parameter adjustment of PID controllers is an old challenge in the field of control system design. Some of methods have been proposed to select the PID coefficients, but they are not completely systematic methods and result in a poorly tuned controller that needs some trail and error. So far, finding new methods to automatically select PID parameters was interest of researches (Almeida et al., 2005). In this chapter the idea of designing optimal PID controller based on genetic algorithms is defined and applied. According to the controller type and location in closed loop control system. Then this idea can be verified and tested by simulation results (see section 5).

4.3.1 Genetic controller algorithm

It is possible to introduce and explain the following computing procedure based on genetic algorithm for optimal selection of controller parameters. This algorithm is clearly shown in Figure 3 and can be explained with the following steps:

1. Specify the controller type and location.
2. Start with a randomly generated population of size ($M_P \times N_P$). (i.e. population of controller parameters (gains) is randomly generated according to a specified parameters range).
3. Calculate the fitness $f(x)$ of each chromosome x in the population.
4. Apply elitism technique to retain one or more best solutions from the population.
5. Apply genetic algorithms operators to generate a new population:
   a. Select a pair of parent chromosomes from the probability of selection being an increasing function of fitness. Selection is done with replacement, which means the same chromosome can be selected more than once to become a parent.
   b. With probability P_C, crossover the pair at a randomly chosen point (chosen with uniform probability) to form two offspring.
   c. Mutate the two offspring at each locus with probability P_m, and place the resulting chromosomes in the new population.
6. Replace the current population with the new population.
7. If stopping criterion is not met then go to step 3 (repeat steps 3-7).
8. Display results and stop program.

To understand this algorithm, we should define the overall system transfer function according to the type and location of the controller. In addition, it is important to determine the following parameters:

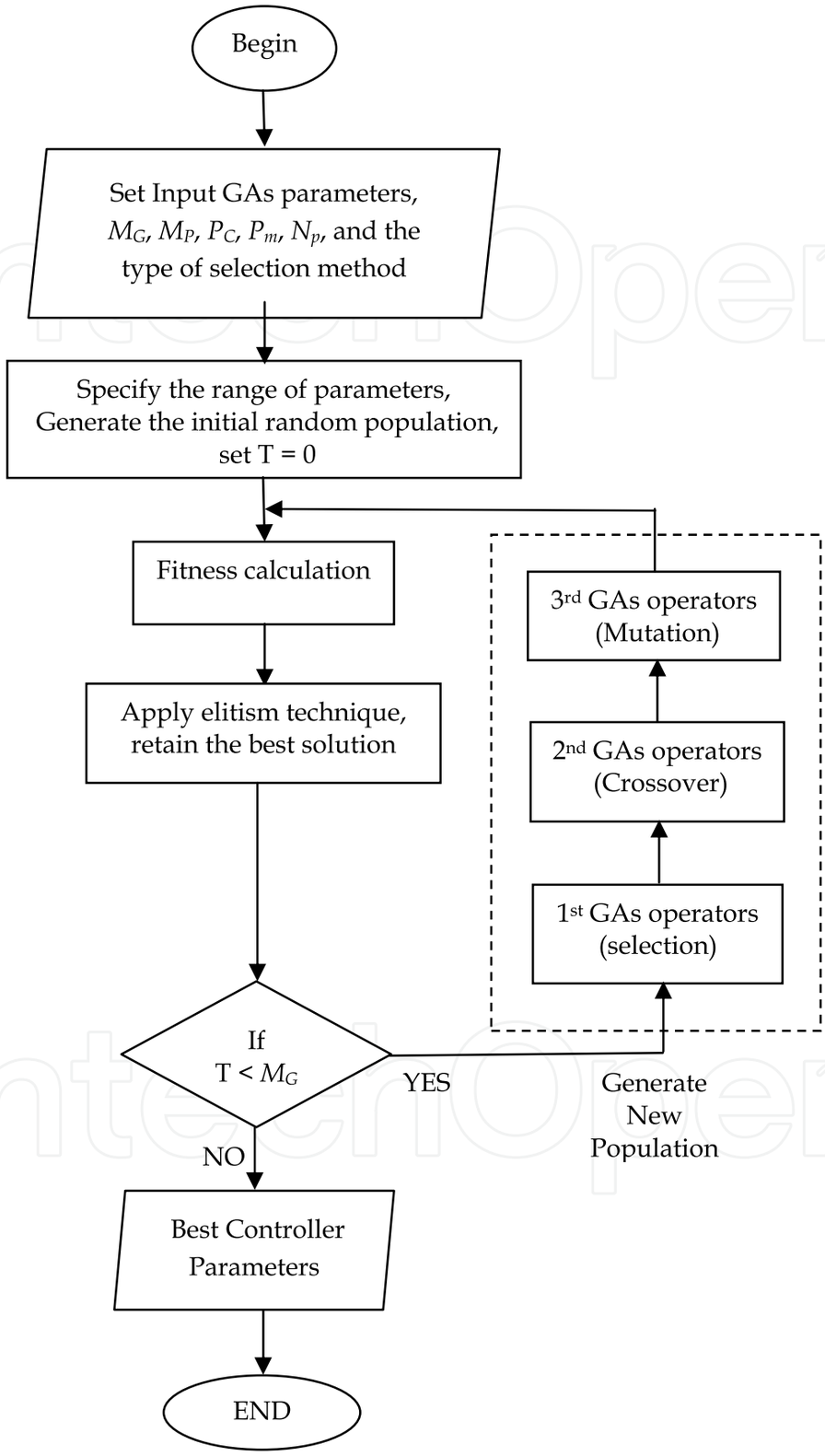


Fig. 3. Flowchart of Genetic PID Controller Algorithm

M_P: Maximum population size.
M_G: Maximum number of generations.
N_P: Number of controller parameters.
R: Range of controller parameters.
P_C: Probability of crossover.
P_m: Probability of mutation.
$f(x)$: The type of fitness function.

The stop criterion may be, for example, maximum number of generations. However, this iterative process leads to the improved performance of candidate set of PID gains. Note that, it is preferable to apply the elitism technique, which is first introduced by Kenneth De Jong in 1975 (Rothlauf, 2006), to forces the genetic algorithms to retain some number of the best individuals at each generation as shown in Fig. 3. Such individuals can be lost if they are not selected to reproduce or if they are destroyed by crossover or mutation. Many researchers have found that elitism significantly improves the genetic algorithms performance (Burns, 2001). Genetic algorithms are typically iterated for anywhere maybe less than 100 or higher. However, each iteration of the process is called a *generation*. The entire set of generations is called *run*. At the end of a run there are often one or more highly fit chromosomes in the population. Since randomness plays a large role in each run, two runs with different random-number seeds will generally produce different detailed behaviors. There are a number of details to fill in, such as the size of the population and the probabilities of crossover and mutation, and the success of the algorithm often depends greatly on these details. In addition the type of selection method used, as well as the type of the fitness function, this is an important issue to achieve optimal solution (Rothlauf, 2006) and (Goldberg, 2002). In control systems the design of PID controllers is difficult and time consuming (Ho, 2003). In this chapter a powerful design method based on real-coded genetic algorithms to solve the minimization of the J_{ITSE} criterion is described. Genetic algorithms provide a much simpler approach to off-line tuning of such controllers than the rather complicated non-genetic optimization algorithms (Burns, 2001). However, in particular PID controllers have many types and different structures, depending on the location of the proportional, integral and derivative control, which can be placed in forward path in cascade with the plant, or in the feedback path.

4.3.2 Tuning genetic PID controller

Figure 4 represent a block diagram of the feedback control system based on genetic PID controller. The controller parameters are tuned by genetic algorithms starting from initial population (Almeida et al., 2005), which is generated randomly and now it is important to calculate the fitness of each chromosome in the population (save best fit individual from the initial population). This is achieved by setting these values to the PID controller, to test the system output response by using unit step input signal as shown in Fig. 4.

4.3.3 Fitness function

Fitness is a numeric value assigned to each member of a population to provide a measure of the appropriateness of each candidate solution (Goldberg, 2002). Fitness functions used in genetic PID controllers are generally based upon the error between the actual and predicted solutions. However, better solutions are achieved according to decreasing the error signal.

The overall operation of a genetic PID controller can be better explained through the flowchart shown in Figure 2. A successful tuning of genetic PID controller can be achieved by choosing a successful fitness function (Rothlauf, 2006). According to Fig. 4 it is possible to collect error signal data to calculate the J_{ITSE} criterion of equation (18), which is desired to be minimized. Whereas this criterion is used to calculate the fitness function, knowing that this function is a maximization function and can take one of the following two forms:

$$f(x) = L_V - \mathrm{J}_{\mathrm{ITSE}}(x) \tag{19}$$

or

$$f(x) = \frac{L_V}{1 + \mathrm{J}_{\mathrm{ITSE}}(x)} \tag{20}$$

Where L_V is a very large value more than 100. The fitness function defined in Equation (20) is much preferable and gives best results; therefore it will be used through the simulation results of this chapter. After that the same procedure will be applied to the next generation. When genetic operators are used to produce offspring better than the parents, continuous improvement of average fitness value from generation to generation is achieved by using the genetic operators. To see the effectiveness of the proposed procedure in this chapter, let us take the following examples.

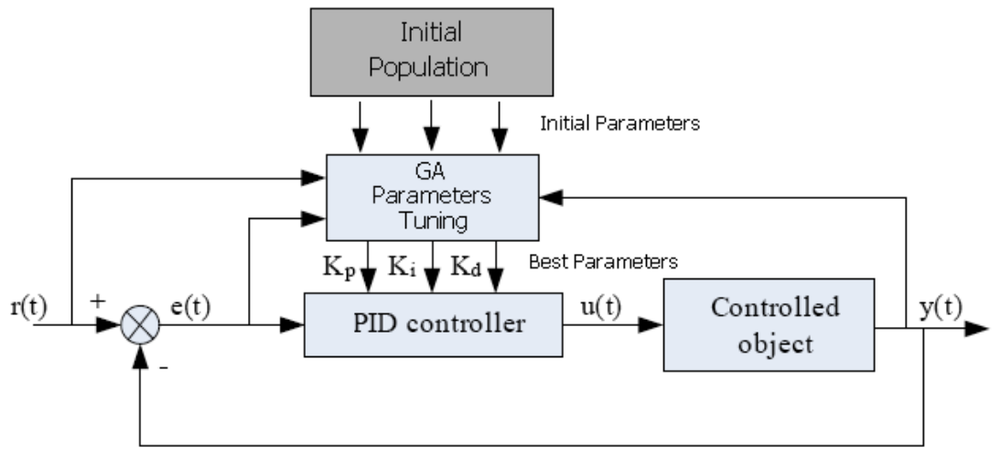


Fig. 4. Tuning genetic PID Controller.

5. Simulation results

5.1 Example 1

Consider the altitude rate control system shown in figure 5 below (Ogata, 2002).

The unit step response of the uncompensated closed loop system is shown in Fig. 6. The response shows high frequency oscillation at the beginning of the response due to the complex poles at -0.0417±j2.4489. The response is dominated by the pole at s=-0.0167, the settling time is at 210 sec. It is desired to speed up the response and also eliminate the oscillatory mode at the beginning of the response.

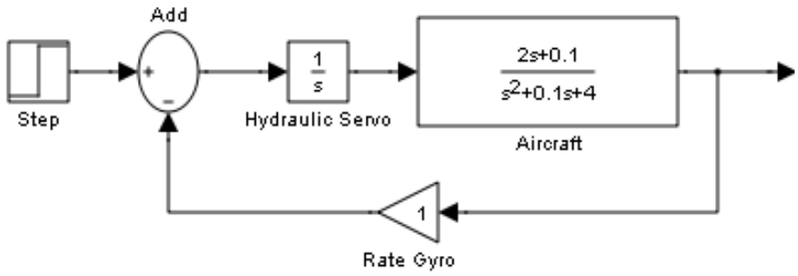


Fig. 5. Block diagram of altitude rate control system

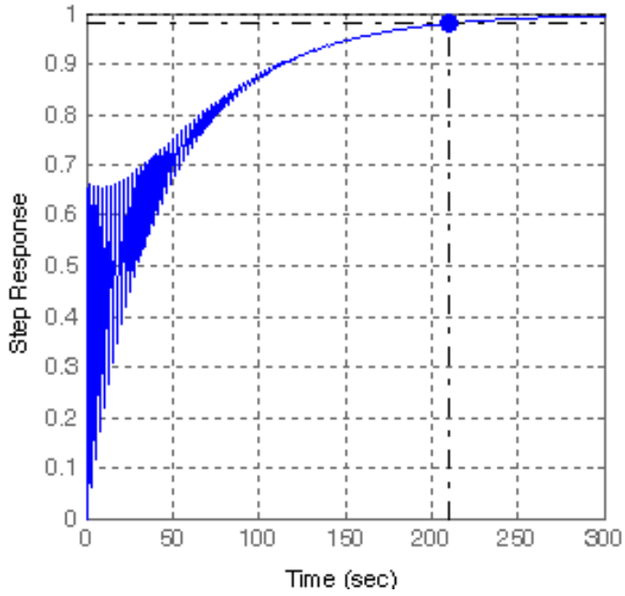


Fig. 6. Step Response of the uncompensated closed loop system, example 1.

Therefore it is desired to improve the system response by using genetic PD controller to enhance the transient response, or using PID controller to enhance both transient and steady state closed loop system response. Let us try to apply the algorithm in section (4.3) step by step to show you the activity of the proposed algorithm. In addition the genetic algorithms which are introduced in this chapter uses roulette wheel selection method and the fitness function defined before (20). In example 1 it is desired to find genetically the parameters of the PD controller, but this need to define first the following genetic input parameters in Table 1.

Genetic Input Parameters						Genetic Output Results	
M_P	M_G	N_P	R	P_C	P_m	K_p	K_d
200	50	2	-500 to 500	0.7	0.09	199.93	67.82

Table 1. Genetic input parameters and results of example 1.

The simulation results of example 1 are shown in Fig. 7 and 8. Figure 7 shows the best and average fitness during generations. Note that an optimal solution is achieved after the 10th

generation as it is clear in Fig. 7 (a). Furthermore Fig. 8(a) shows the closed loop step response with genetic PD controller of the system defined in example 1. it is clear that the system response is improved and achieved better performance by using genetic PD controller. Figure 8(b) shows the control signal of the system. Finally Fig. 8(c) and (d) illustrate the variations of the best values of the controller gains K_d, and K_p respectively in each generation.

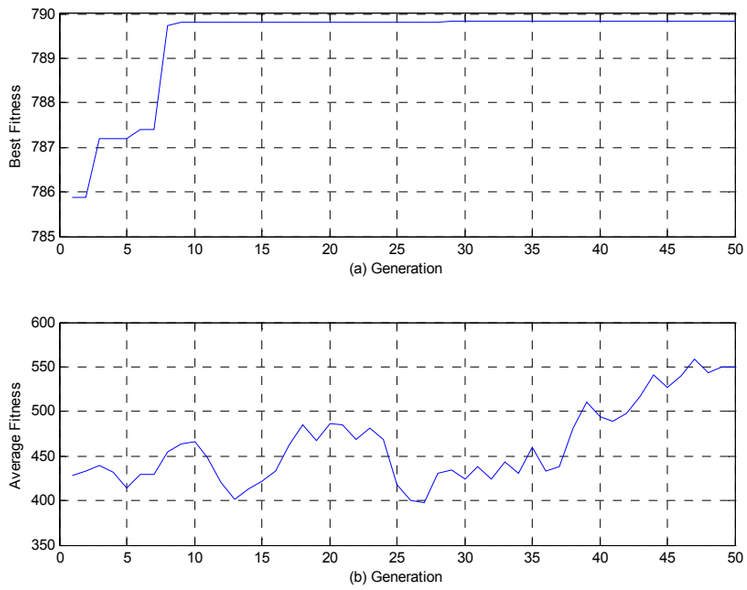


Fig. 7. Variation of fitness during generations. (a) Best fitness, (b) Average fitness.

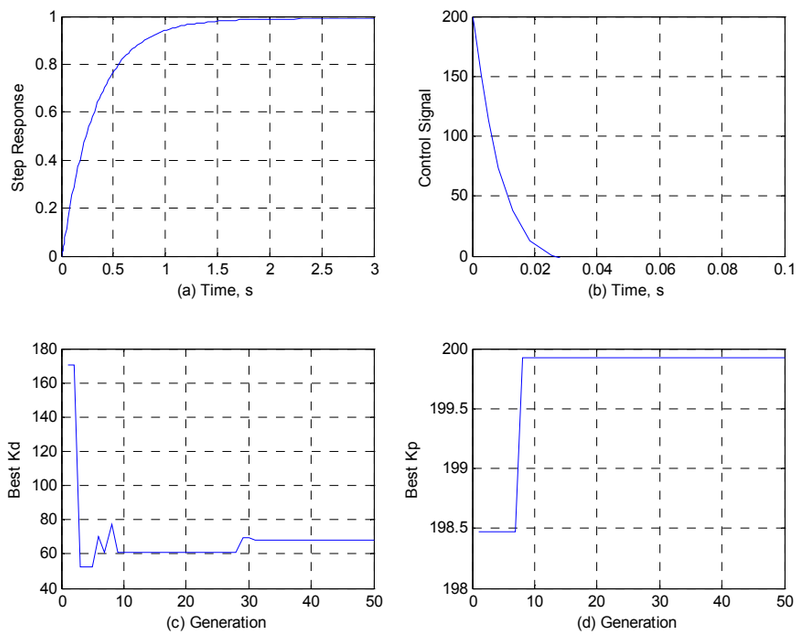


Fig. 8. Simulation results of example 1. (a) Step response of closed loop system with genetic PD controller, (b) Control input, $u(t)$, (c) variation of best value K_d in each generation, (d) variation of best value K_p in each generation.

5.2 Mathematical model of DC motor

Consider the separately excited DC motor system with armature control (Namazov and Basturk , 2010) and (Nasri et al., 2007), the voltage applied to the armature of the motor is adjusted without changing the voltage applied to the field. Figure 9 shows a separately excited DC motor equivalent model. This model is then built in using MATLAB/Simulink as illustrated in Fig. 10.

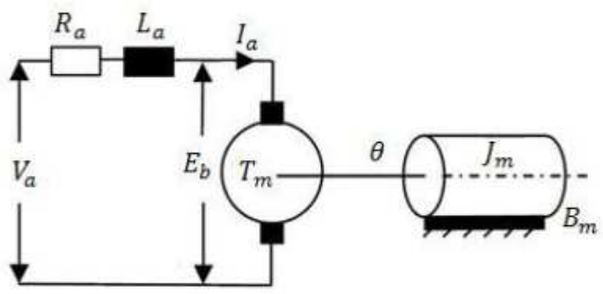


Fig. 9. Separately excited DC motor.

$$V_a(t) = R_a.i_a(t) + L_a.\frac{di_a(t)}{dt} + e_b(t) \tag{21}$$

$$e_b(t) = K_b.\omega(t) \tag{22}$$

$$T_m(t) = K_m.i_a(t) \tag{23}$$

$$T_m(t) = J_m.\frac{d\omega(t)}{dt} + B_m.\omega(t) \tag{24}$$

where
$V_a(t)$ = armature voltage (V)
R_a = armature resistance (Ω)
L_a = armature inductance (H)
I_a = armature current (A)
E_b = back emf (V)
w = angular speed (rad/s)
T_m = motor torque (N m)
θ = angular position of rotor shaft (rad)
J_m = rotor inertia (kg m2)
B_m = viscous friction coefficient (Nm s/rad)
K_m = motor torque constant (Nm/A)
K_b = back emf constant (V s/rad)

Let us combine the upper equations (21) to (24) together and taking the Laplace transforms yields,

$$V_a(s) = \omega(s).\frac{1}{K_m}.[L_a.J_m.s^2 + (R_a.J_m + L_a.B_m).s + (R_a.B_m + K_b.K_m)] \tag{25}$$

Then the relation between rotor shaft speed and applied armature voltage is represented by transfer function:

$$\frac{\omega(s)}{V_a(s)} = \frac{K_m}{L_a.J_m.s^2 + (R_a.J_m + L_a.B_m).s + (R_a.B_m + K_b.K_T)}. \tag{26}$$

And the transfer functions between shaft position and armature voltage at no-load are:

$$\frac{\theta(s)}{V_a(s)} = \frac{K_m}{L_a.J_m.s^3 + (R_a.J_m + L_a.B_m).s^2 + (R_a.B_m + K_b.K_T).s}. \tag{27}$$

where

$$\theta(s) = \frac{1}{s}\omega(s) \tag{28}$$

Therefore, now it is possible to build the Simulink DC motor model, using the transfer functions (25) to (28) as shown in Fig. 10 below. Motor model was converted to a 2-in 4-out subsystem. Input ports are armature voltage and load torque (T_{load}) and the output ports are armature current, motor torque, angular shaft speed and position.

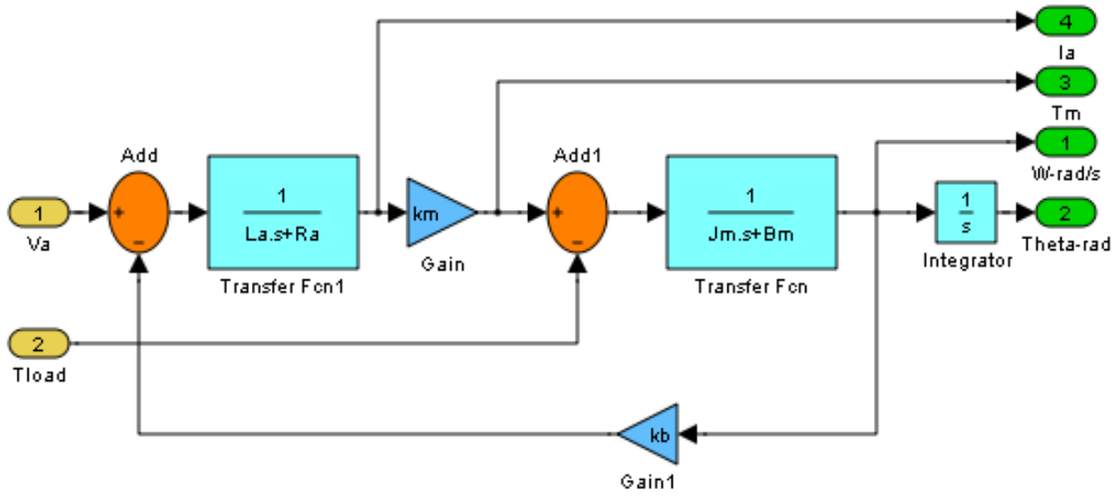


Fig. 10. DC motor Simulink model.

5.2.1 Example 2

Consider A 3.70 kW, 240V, 1750 rpm DC motor with the following parameters in Table 2 (Namazov and Basturk , 2010):

Ra (Ω)	La (H)	J_m (kg m²)	B_m (Nms/rad)	K_m (Nm / A)	K_b (Vs/rad)
11.2	0.1215	0.02215	0.002953	1.28	1.28

Table 2. DC motor parameters.

The closed loop output response of the uncompensated DC motor speed control system is shown in Fig. 11. However it is clear that the system is stable but it has a large steady state error.

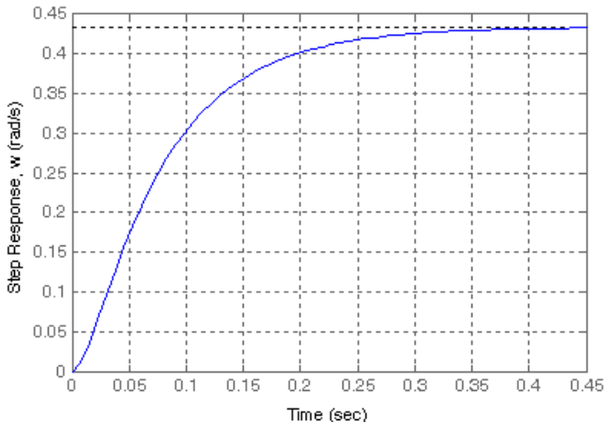


Fig. 11. The closed loop output response of the uncompensated DC motor speed control system.

Genetic PID controller

To improve the closed loop output response of the DC motor speed control system. It is desired to use the genetic PID controller as shown in Fig. 12. With the following genetic parameters in Table 3

Genetic Input Parameters						Genetic Output Results		
M_P	M_G	N_P	R	P_C	P_m	K_p	K_d	K_i
500	20	3	-500 to 500	0.75	0.09	3.8087	0.0203	9.93

Table 3. Genetic input parameters and results of example 2.

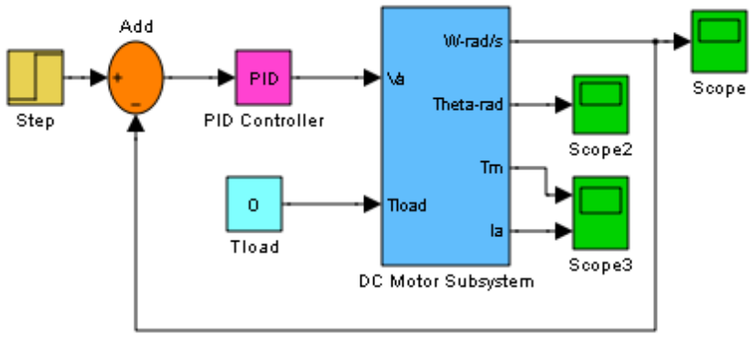


Fig. 12. Speed Control of DC Motor System with PID Controller.

The simulation results of example 2 are shown in Fig. 13 to 15. Figure 13 shows the best and average fitness during generations. Note that an optimal solution is achieved around the 10th generation as it is clear in Fig. 13 (a). Furthermore Fig. 13(a) shows the closed loop step response with genetic PID controller of the speed control system. It is clear that the system

response is improved and achieved better performance by using the proposed genetic PID controller. Figures 14(b) to (d) illustrates the variations of the best values of the controller gains K_d, K_p and K_i respectively in each generation. In addition Fig. 15 shows the Root locus plot and Bode plot of the open loop system of example 2 with PID controller.

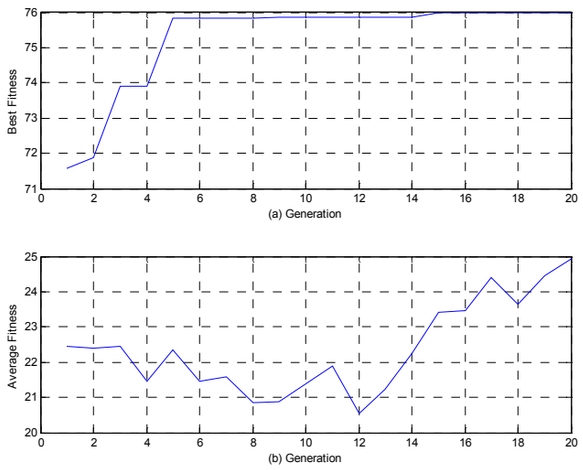


Fig. 13. Variation of fitness during generations of example 2. (a) Best fitness, (b) Average fitness.

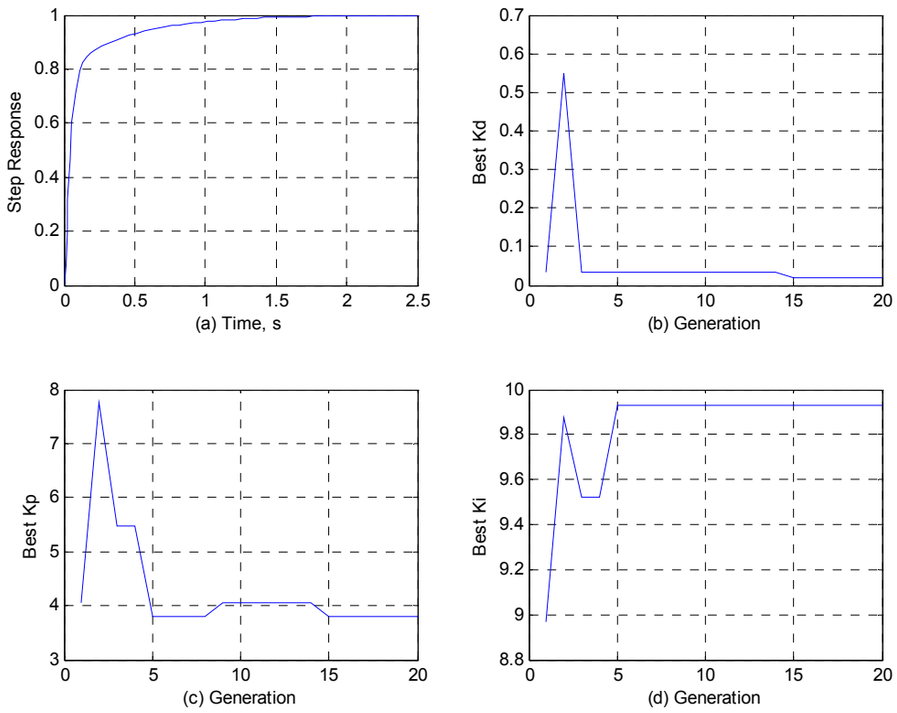
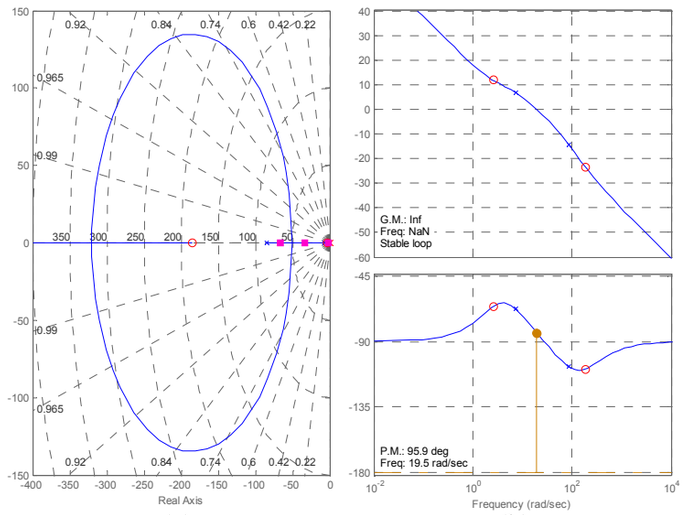


Fig. 14. Simulation results of example 2 at no-load condition. (a) Step response of speed control system with genetic PID controller, (b) variation of best value K_d in each generation, (c) variation of best value K_p in each generation, (d) variation of best value K_i in each generation.



(a) (b)

Fig. 15. Open loop test of example 2 with PID controller (a) Root locus plot, (b) Bode plot.

From table 3, it is clear that the derivative gain Kd has a small value. This is because we need to improve the transient response of the system or to reduce the steady state error. Therefore we can use PI controller to improve such system response. Moreover, to test the system of example 2 under load condition, with the same results obtained previously in Table 3. A constant load torque of 5 N.m at time 2 second is applied to the closed loop system as shown in Fig. 16. The simulation results show the step response of speed control system with genetic PID controller under load test condition. In addition the motor torque and Armature current of the motor are illustrated in Fig. 16 (b) and (c) respectively.

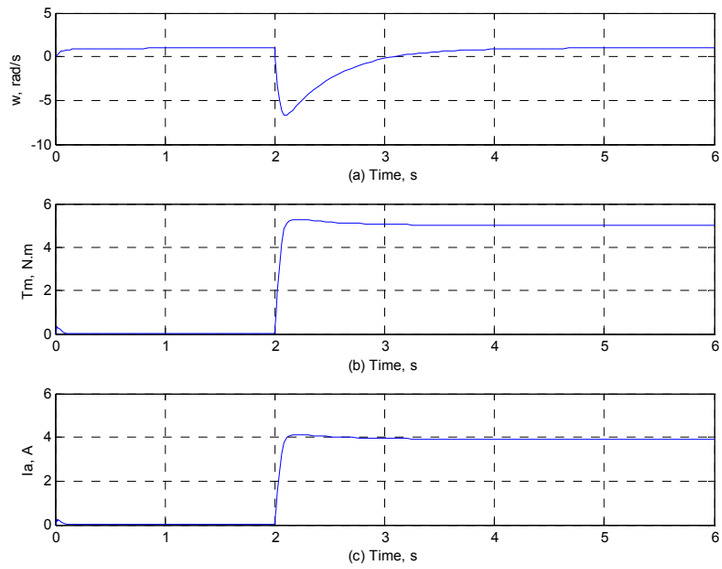


Fig. 16. Simulation results of example 2 with constant T_{load} of 5 N.m at time 2 sec.
(a) Step response of speed control system with genetic PID controller, (b) Motor Torque,
(c) Armature current of the motor.

5.2.2 Example 3

For the Position control shown in Fig. 17 below. It is desired to achieve optimal output response by applying the algorithm proposed in this chapter. The genetic PID controller shown in Fig. 12 is used to improve the output system response with the following genetic parameters in Table 4.

Genetic Input Parameters						Genetic Output Results		
M_P	M_G	N_P	R	P_C	P_m	K_p	K_d	K_i
500	20	3	-500 to 500	0.75	0.09	9.7058	4.0305	0.050714

Table 4. Genetic input parameters and results of example 3 with PID controller.

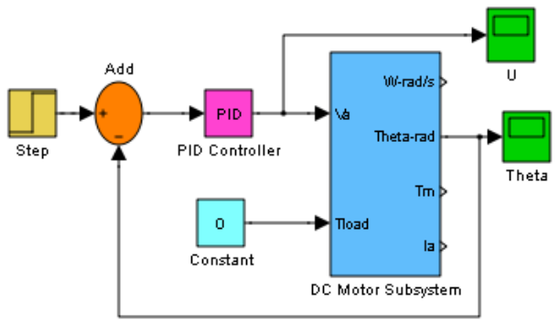


Fig. 17. DC motor position control system with genetic PID controller.

The simulation results of example 3 at no-load condition with genetic PID controller are shown in Fig. 18 below.

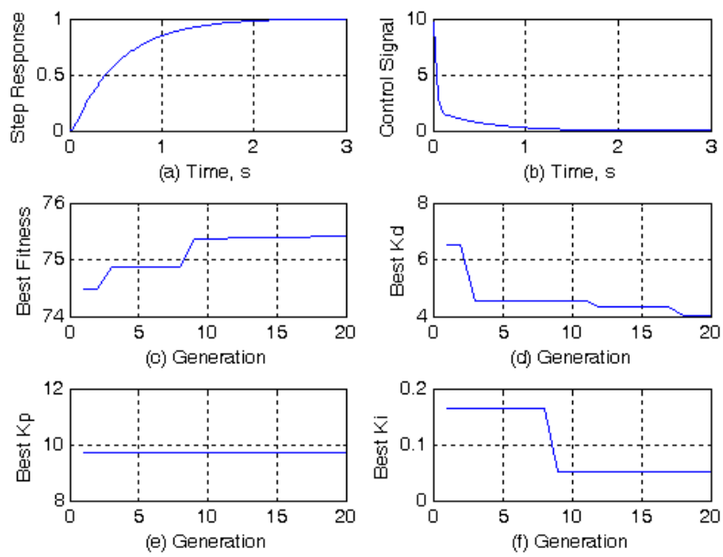


Fig. 18. Simulation results of example 3 at no-load condition with genetic PID controller. (a) Step response of position control system, (b) Control input, (c) Variation of best fitness during generations, (d) variation of best value K_d in each generation, (e) variation of best value K_p in each generation, (f) variation of best value K_i in each generation.

Note that the value of the integral gain is very small, so it can be neglected. In addition it is possible to achieve system stability by using only PD controller as shown in Fig. 19 below. Table 5 represents the genetic input parameters and output results of example 3 with PD controller.

Genetic Input Parameters						Genetic Output Results	
M_P	M_G	N_P	R	P_C	P_m	K_p	K_d
200	50	2	-500 to 500	0.8	0.05	19.91	6.52

Table 5. Genetic parameters and results of example 3 with PD controller.

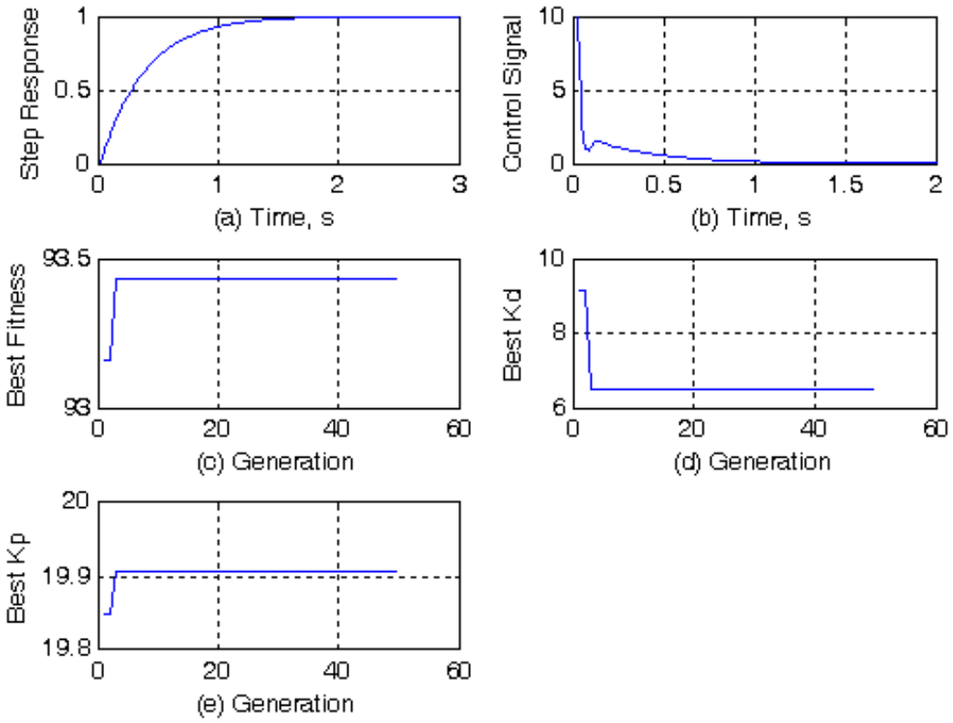


Fig. 19. Simulation results of example 3 at no-load condition with genetic PD controller. (a) Step response of position control system, (b) Control input, (c) Variation of best fitness during generations, (d) variation of best value K_d in each generation, (e) variation of best value K_p in each generation.

6. Robust stability using genetic PID controller

The principal objective of control engineering is to design control systems which are robust with respect to external disturbances and modelling uncertainty. The attenuation of disturbances will be discussed in this section clearly. This is done by adding two types of disturbances in order to show you the activity of genetic controllers against disturbances. Figure 20 represents the closed loop system with genetic controller, which is designed according to the algorithm proposed in this chapter.

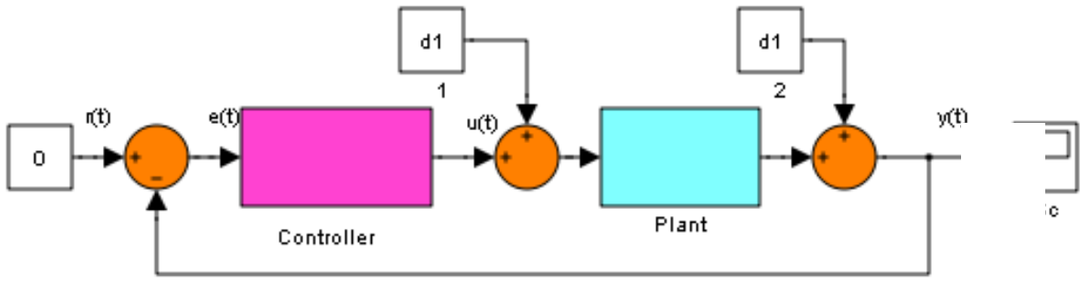


Fig. 20. Closed loop system test against input and output disturbances.

The disturbance can affect plants and no countermeasure can be applied, as the controller can only counteract after the changes at the system output (Bartoszewicz, 2011) and (Mansour, 2011).

Referring to example 1 (section 5.1), by using the same results of genetic controller obtained before. In order to test the system performance against input and output disturbances, this is achieved by assuming zero reference input and taking the effects of each disturbance alone input disturbance (d1) and output disturbance (d2), which are assumed to be a constant value applied to the system. Figure 21 and 22 shows the closed loop output response subjected to input disturbance and output disturbance respectively. The results show the viability of the system against disturbances. In addition this means that the genetic controller give an optimal results that achieve robust stability.

The same procedure is applied to example 2 for different load cases and for the two types of disturbances as shown in figures 23 to 25. The results are clearly seam to be effective against disturbances rejection.

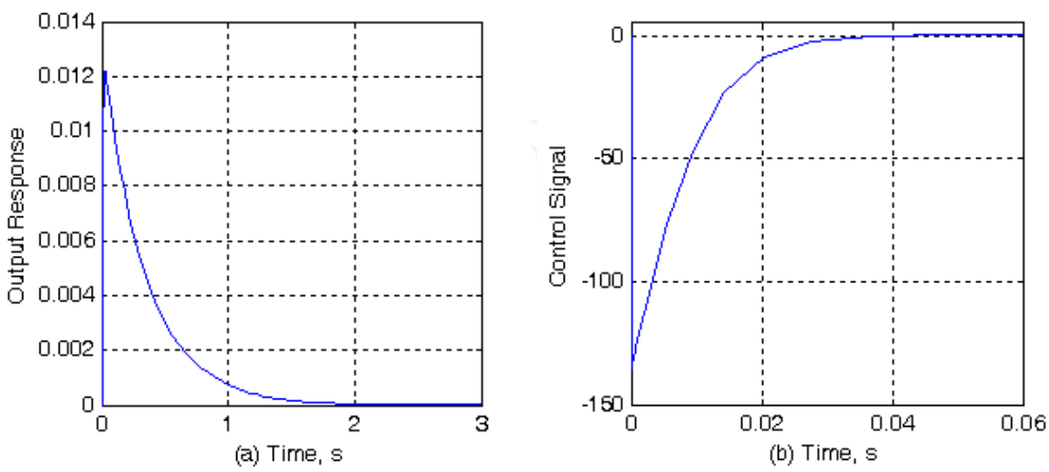


Fig. 21. Output response of closed-loop system of example 1against unit step load disturbance (at the plant input). (a) System output response, (b) Control input.

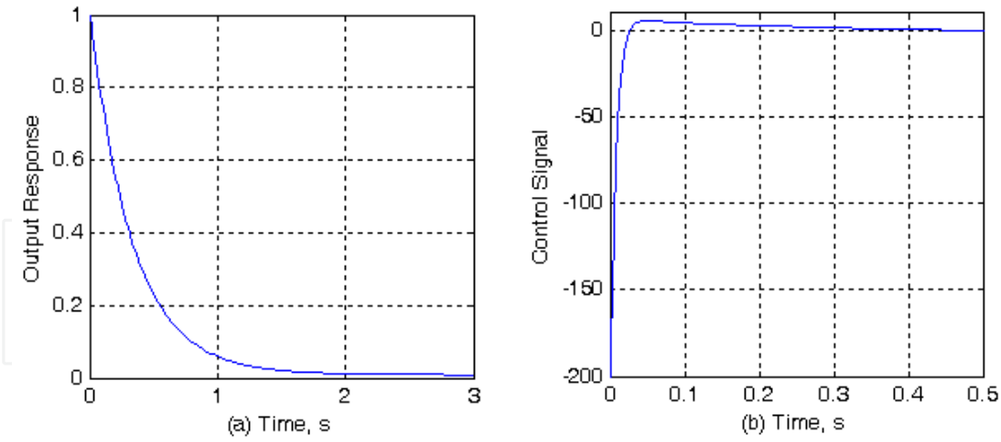


Fig. 22. Output response of closed-loop system of example 1against unit step output disturbance. (a) System output response, (b) Control input.

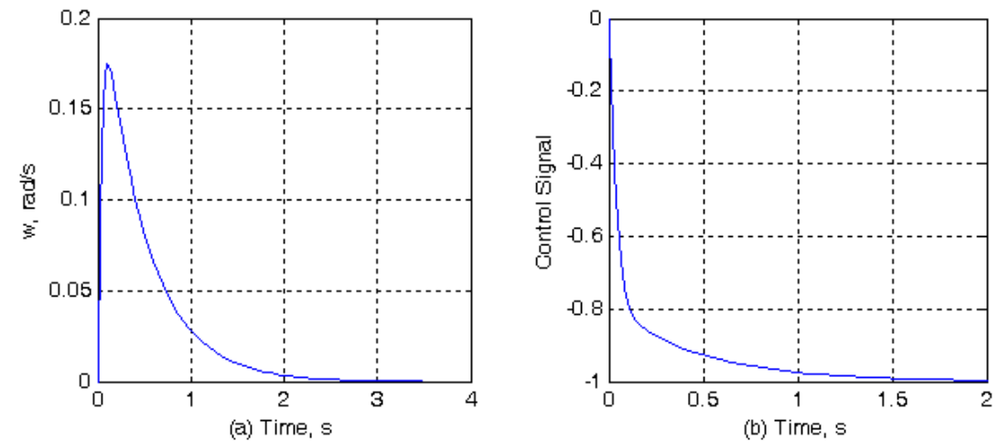


Fig. 23. Output response of closed-loop system of example 2 at no-load against unit step load disturbance. (a) System output response, (b) Control input.

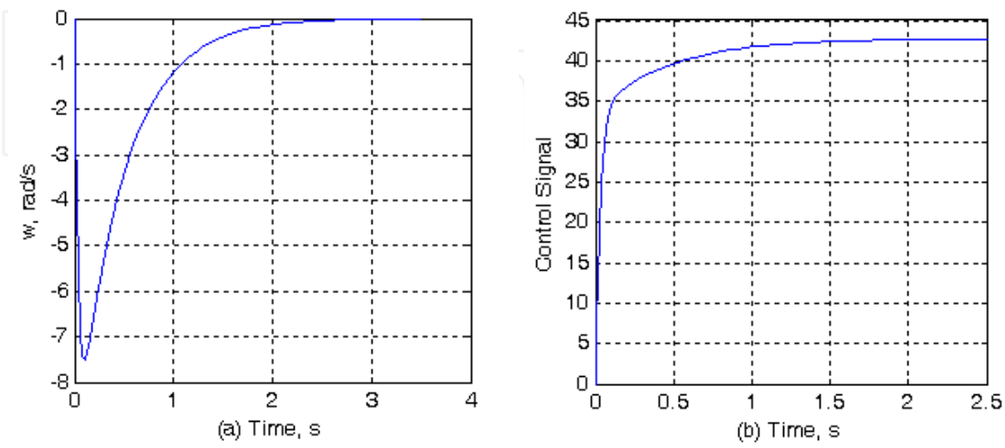


Fig. 24. Output response of closed-loop system of example 2, T_{load}= 5 N.m against unit step load disturbance. (a) System output response, (b) Control input.

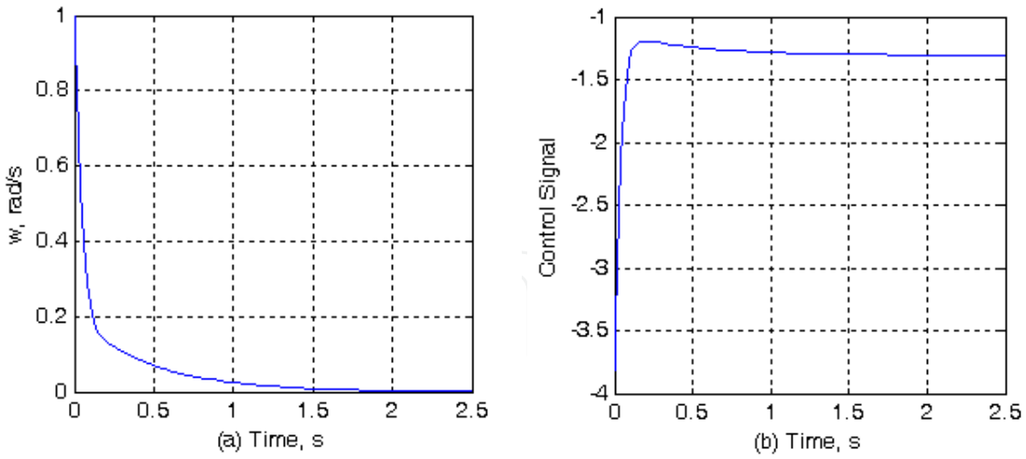


Fig. 25. Output response of closed-loop system of example 2 at no-load against unit step output disturbance. (a) System output response, (b) Control input.

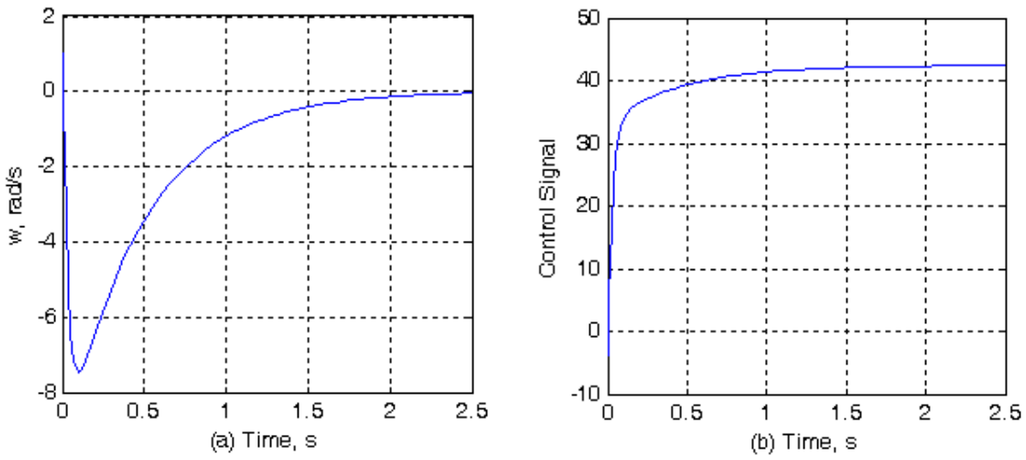


Fig. 26. Output response of closed-loop system of example 2, with T_{load} = 5 N.m against unit step output disturbance. (a) System output response, (b) Control input.

7. Conclusions

PID controllers are widely used in industrial control applications due to their simple structures, comprehensible control algorithms and low costs. In this chapter a new design method to determine PID controller parameters using genetic algorithms is presented. The aim of this chapter is to achieve parameters tuning of a PID controller using genetic algorithms. To reach that goal a fitness function in terms of system error signal which are functions of performance criteria is introduced. This is an important factor to achieve optimal controller parameters which give satisfied results. Maximization of such a fitness function by genetic algorithms causes a satisfactory steady state error and maximum over shoot as well as less control energy in comparison with conventional control methods.

In this work different control systems are tested through simulation results by using MATLAB/SIMULINK program to show that the proposed controller can perform an

efficient search for the optimal PID controller, which tend to improve the dynamic performance of the system in a better and simple way. In addition the control system is tested against external disturbances. However robustness is a significant property that allows for the control system to maintain its function despite of external disturbances. For future work, it is possible to apply the algorithm proposed in this chapter to many different types of controllers for both SISO and MIMO systems. Also it can be applied efficiently to discrete types PID controllers.

8. Acknowledgements

First of all, the author would like to thank the government of Bahrain by the Ministry of Education for their continued support for scientific research and to attain the highest levels of the academic field. Furthermore, my deepest gratitude goes to my parents, wife and sons for their unconditional support, understanding, patience and encouragement. It is to them that this work is dedicated.

9. References

A. D. Rodić (2009) *Automation and Control Theory and Practice*, Published by InTech.

Andrzej Bartoszewicz (2011) *Robust Control Theory and Applications*, Published by InTech

David E. Goldberg (2002) *Genetic Algorithms in search, Optimization, and Machine learning* 5th edition Pearson Education Asia.

Franz Rothlauf (2006) *Representations for Genetic and Evolutionary Algorithms*, Springer-Verlag Berlin Heidelberg.

Gustavo Maia de Almeida, Valceres Vieira Rocha e Silva, Erivelton Geraldo Nepomuceno, and Ryuichi Yokoyama (2005), Application of Genetic Programming for Fine Tuning PID Controller Parameters Designed Through Ziegler-Nichols Technique, ICNC 2005, LNCS 3612, pages 313–322, *Springer*-Verlag Berlin Heidelberg.

Katsuhiko Ogata (2002) *Modern Control Engineering* 4th edition Prentice Hall.

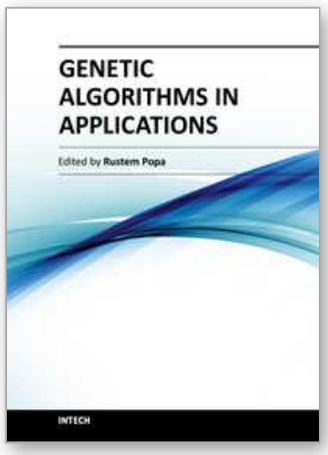
Manafeddin Namazov and Onur Basturk (2010) DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods An Official Journal of Turkish Fuzzy Systems Association Vol.1, No.1, pages 36-54, 2010.

Mehdi Nasri, Hossein Nezamabadi-pour, and Malihe Maghfoori (2007) A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor, *World Academy of Science, Engineering and Technology* 26, pages 211-215.

Ming-Tzu Ho (2003) Synthesis of H^{∞} PID controllers: A parametric approach, Available online at www.sciencedirect.com, PERGAMON, *Automatica*, Vol. 39, pages 1069 – 1075.

Roland S. Burns (2001) *Advanced Control Engineering*, Butterworth-Heinemann, Linacre House, Jordan Hill, Oxford.

Tamer Mansour (2011) *PID Control, Implementation and Tuning* Published by InTech



Genetic Algorithms in Applications

Edited by Dr. Rustem Popa

ISBN 978-953-51-0400-1
Hard cover, 328 pages
Publisher InTech
Published online 21, March, 2012
Published in print edition March, 2012

Genetic Algorithms (GAs) are one of several techniques in the family of Evolutionary Algorithms - algorithms that search for solutions to optimization problems by "evolving" better and better solutions. Genetic Algorithms have been applied in science, engineering, business and social sciences. This book consists of 16 chapters organized into five sections. The first section deals with some applications in automatic control, the second section contains several applications in scheduling of resources, and the third section introduces some applications in electrical and electronics engineering. The next section illustrates some examples of character recognition and multi-criteria classification, and the last one deals with trading systems. These evolutionary techniques may be useful to engineers and scientists in various fields of specialization, who need some optimization techniques in their work and who may be using Genetic Algorithms in their applications for the first time. These applications may be useful to many other people who are getting familiar with the subject of Genetic Algorithms.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Osama Y. Mahmood Al-Rawi (2012). Enhancing Control Systems Response Using Genetic PID Controllers, Genetic Algorithms in Applications, Dr. Rustem Popa (Ed.), ISBN: 978-953-51-0400-1, InTech, Available from: http://www.intechopen.com/books/genetic-algorithms-in-applications/enhancing-control-systems-response-using-genetic-pid-controllers

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen