

# Gravitational Search Algorithm Optimization for PID Controller Tuning in Waste-water Treatment Process

Mohamad Saiful Islam Aziz<sup>a</sup>, Sophan Wahyudi Nawawi<sup>a\*</sup>, Shahdan Sudin<sup>a</sup>, Norhaliza Abdul Wahab<sup>a</sup>, Mahdi Faramarzi<sup>a</sup>, Masdinah Alauyah Md. Yusof<sup>b</sup>

<sup>a</sup>Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

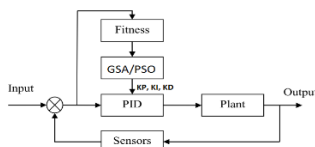
<sup>b</sup>Language Academy, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

\*Corresponding author: sophan@fke.utm.my

## Article history

Received :15 August 2014  
Received in revised form :  
5 January 2015  
Accepted :10 February 2015

## Graphical abstract



## Abstract

This paper presents a new approach of optimization technique in the controller parameter tuning for waste-water treatment process (WWTP) application. In the case study of WWTP, PID controller is used to control substrate (S) and dissolved oxygen (DO) concentration level. Too many parameters that need to be controlled make the system becomes complicated. Gravitational Search Algorithm (GSA) is used as the main method for PID controller tuning process. GSA is based on Newton's Law of Gravity and mass interaction. In this algorithm, the searcher agents survey the masses that interact with each other using law of gravity and law of motion. For WWTP system, the activated sludge reactor is used and this system is multi-input multi-output (MIMO) process. MATLAB is used as the platform to perform the simulation, where this optimization is compared to other established optimization method such as the Particle Swarm Optimization (PSO) to determine whether GSA has better features compared to PSO or vice-versa. Based on this case-study, the results show that transient response of GSA-PID was 20%-30% better compared to transient response of the PSO-PID controller.

**Keywords:** Gravitational Search Algorithm (GSA); optimization; Particle Swarm Optimization (PSO); PID controller; waste-water treatment process (WWTP)

© 2015 Penerbit UTM Press. All rights reserved.

## 1.0 INTRODUCTION

The control design of waste-water treatment plant (WWTP) becomes very important nowadays due to the changes in parameters and influent characteristics. WWTP involves a multi-variable process which is highly complex thus tuning of the controller is not an easy job. The waste-water treatment plant (WWTP) is a process of treating the water by removing the organic waste and nutrients[1]. This process can be categorized into several stages which are primary, secondary, and tertiary treatments. Activated sludge process (ASP) is one of the processes that fall under the secondary treatment. ASP is a biological process which is responsible for maintaining the pollutant substrate and dissolving oxygen within an acceptable range. This process involves a number of interacting controls. After the commissioning of the plant, the controller parameters are left unchanged. However, due to the environment conditions (e.g. rain and flood) poor plant performance can be observed.

A PID (Proportional-Integral-Derivative) controller is a common feedback loop component that is widely used in the industrial control system. PID controller has been commercially used due to its simple structure and robust performance in a wide range of operating conditions[2]. Over the years, lots of PID tuning method have been proposed such as Ziegler-Nichols (Z-N),

Frequency-domain method and Time-domain method. With the developments of Artificial Intelligent (AI), many intellectual algorithms have been proposed for PID tuning purpose such as Ant Colony Optimization (ACO), Bee Algorithm, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The PSO and GA are widely used for PID controller optimization comparison. In[3], [4],[5] shows that PSO is superior to GA because PSO converges with less number of functional evaluations, is consistent for simple system and in general ITAE is preferable for quick settling time. In [6], it is shown that PSO is superior to GA for liquid level tank system.

Gravitational Search Algorithm (GSA) is one of the latest intellectual algorithms that falls under the artificial intelligent field. GSA is developed by Rashedi *et.al*[7] where this algorithm is based on Newton's law of gravity and law of motion. In the purposed algorithm, agents are considered as particles and their performance are based on its masses. All the particles are attached to each other by a force called gravitational force and this force causes the global movement of all particles to particle with heavier masses. The particle with heavier masses will correspond to good solutions and move more slowly and on the other hands particle with light masses correspond to poor solutions and moved faster. Therefore, particle with lighter mass tends to move towards particle with heavy mass much faster. This guarantees the exploitation step of the algorithm.

In this paper, GSA is used as the main tuning method for the PID control controller for the closed loop of WWTP system. The performance of the GSA-PID controller will be compared with the performance results of PID tuning by PSO method.

### 2.0 MATHEMATICAL MODELLING OF WWTP

The biological process is the most popular method in waste-water treatment as the most important part of waste-water treatment is the removal process of organic matter that dissolved in waste-water. This removal of organic matter using biological processes is an aerobic process. The aerobic process for this plant takes place in aeration tank, where the waste-water is aerated with oxygen using an activated sludge. Mathematical modelling system for waste-water treatment process can be described by a transfer function matrix, as visualized in (1) below,

$$Y(s) = G(s)U(s) \tag{1}$$

where  $Y(s)$ ,  $G(s)$  and  $U(s)$  represent the output, process plant and input respectively. The waste-water treatment plant that will be used for this research is shown in Figure 1 below. This plant is comprised of an aerated tank and clarifier. There are four inputs and four outputs but for the purpose of this study, only two inputs: dilution rate, air flow rate and two outputs: substrate and dissolved oxygen will be considered[8].

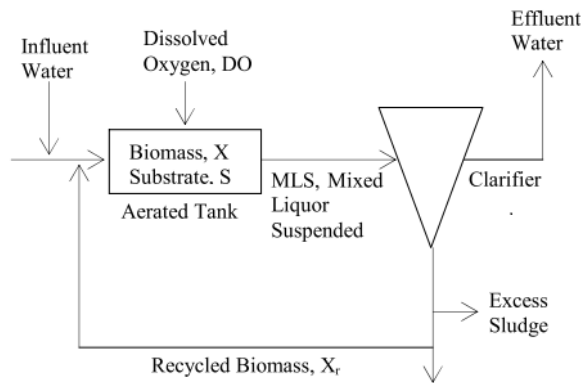


Figure 1 Schematic of waste-water treatment plant (WWTP)

The behaviours of the waste-water treatment plant can be described by four different nonlinear equations as expressed in Equations (2) until (5) below:

$$\dot{X}(t) = \mu(t)X(t) - D(t)(1+r)X(t) + rD(t)X_r(t) \tag{2}$$

$$\dot{S}(t) = -\frac{\mu(t)}{Y}X(t) - D(t)(1+r)S(t) + D(t)S_{in} \tag{3}$$

$$\dot{C}(t) = -\frac{K_o\mu(t)}{Y}X(t) - D(t)(1+r)C(t) + K_{LA}(C_s - C(t)) + D(t)C_{in} \tag{4}$$

$$\dot{X}_r(t) = D(t)(1+r)X(t) - D(t)(\beta+r)X_r(t) \tag{5}$$

Where the state variables  $\dot{X}(t)$ ,  $\dot{S}(t)$ ,  $\dot{C}(t)$  and  $\dot{X}_r(t)$  represent the concentrations of biomass, substrate, dissolved oxygen and recycled biomass respectively.  $D(t)$ ,  $\mu(t)$ ,  $S_{in}(t)$ ,  $C_{in}(t)$ ,  $C_s$ ,  $Y$ ,  $K_o$ ,  $K_{LA}$ ,  $r$  and  $\beta$  represent the dilution rate, specific growth rate, substrate concentrations influent steams, dissolved oxygen concentration of influent steams, constant of maximum dissolved

oxygen, rate of microorganism growth, model constant, constant of oxygen transfer rate coefficient, ratio of recycled and ratio of waste flow to the influent flow rate respectively.

### 3.0 PRINCIPLE OF ALGORITHM

#### A. Particle Swarm Optimization (PSO)

The PSO is introduced by Kennedy and Eberhart in 1995, and it has become one of the favorites in optimization algorithm solutions [9]. PSO is based on the movements of the group behavior such as bird flocking and fish schooling. An improvement of PSO is done in 1998 by the introduction of inertia weight into PSO by Shia and Eberhart[10]. To provide a balance between global and local explorations, a suitable selection of inertia weight must be done. By doing that, algorithm convergence can be controlled and the best value of fitness function can be found[10].

In PSO, individuals are called as particles and these particles are “evolved” by the cooperation and competition among themselves through generations. A particle represents a potential solution to a problem. Each particle adjusts its flying according to its own flying experience and its companion flying experience. By this movement, its particle can be said to have the velocity. Each particle is trying to find the optimum solution in the solution space and this value is called as personal best or ‘*pbest*’. Any particle in the neighborhood also can provide another best value and this value is called global best or ‘*gbest*’. In other words, PSO concept lies in the movement of each particle toward these *pbest* and *gbest* locations, with random acceleration at each time.

To modify its positions and velocity in the search space, information related to particle such as current position, current velocity, distance between current position and *pbest* position, and distance between current position and *gbest* position will be used. The modified new position and new velocity of each particle can be calculated from this informations as shown in following formulas:

$$V_{id}^{k+1} = wV_{id}^k + c_1rand_1() (pbest_{id} - x_{id}^k) + c_2rand_2() (gbest_{id} - x_{id}^k) \tag{6}$$

$$X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1} \tag{7}$$

Where:

- $i$  = Pointer of iteration (generations)
- $d$  = Number of dimension
- $k$  = Pointer of iteration (generations)
- $V_{id}^k$  = Velocity of the  $i$ th particle at  $n$ th dimension
- $w$  = Inertia weight factor
- $c_1, c_2$  = Acceleration constant
- $rand_1, rand_2$  = Random number between 0 and 1
- $X_{id}^k$  = Current position of the  $i$ th particle at  $n$ th dimension
- $pbest_i$  = *pbest* of the  $i$ th particle at  $n$ th dimension
- $gbest_i$  = *gbest* of the  $i$ th particle at  $n$ th dimension

In this analysis, the parameter specific for benchmarking the PSO-PID controller were set to  $i = 100$ ,  $d = 3$  which represent PID controller parameters ( $K_p$ ,  $K_i$ ,  $K_D$ ),  $C_1 = 0.9$ ,  $C_2 = 0.4$ . The range of PID controller parameters were selected from 0 to 100. This range was selected based on standardization value with other researchers about the PID controller application [11], [12]. From the procedure above, the maximum velocity,  $V_{max}$  determined the

resolution or fitness. Regions were searched between the present position and the target positions. If  $V_{max}$  was too high, particle would have moved past good solutions. If  $V_{max}$  was too small, the convergence would have been slower and would have led to an inaccurate searching. Based on the experiences with PSO,  $V_{max}$  was often between 10%-25% of the dynamic range of the velocity.

The following weighting function is usually utilized in velocity update function;

$$w = w_{max} - \frac{(w_{max} - w_{min}) * iter}{max\ iter} \tag{8}$$

Where:

- $w_{max}$  = Initial weight
- $w_{min}$  = Final weight
- $iter$  = Current iteration number
- $max\ iter$  = Maximum iteration number

**B. Gravitational Search Algorithm (GSA)**

The Gravitational Search Algorithm (GSA) is one of the latest stochastic search algorithm introduced by Rashedi *et.al* in 2009[7]. This algorithm is based on the two Newtonian laws: Law of Gravity and Law of Motion. In this algorithm, the agents are taken into consideration as particle and their performances are monitored based on their masses. By far, every particle represents a solution or can be part of the solution to the problem. All the particles are attached to each other by a force called gravitational force and this force causes the global movement of all particles towards particle with heavier masses. The heavier masses will correspond to good solutions and move more slowly and conversely light mass particles resembling the poor solutions. Based on that, light mass particles tends to move towards heavy mass particles much faster.

In GSA, each mass can be categorized into four specifications: position, active gravitational mass ( $M_{ai}$ ), passive gravitational mass ( $M_{pi}$ ), and inertia mass ( $M_{ii}$ ). The algorithm will properly adjusted its gravitational and inertia masses until by lapse time, that particle will be attracted to the particle with heaviest mass [7], [13].

For the starting of the algorithm, the position of a system can be described as follows:

$$X_i = (x_i^1 \dots x_i^d \dots x_i^n) \text{ for } i = 1, 2, \dots, N \tag{9}$$

where  $n$  is the space dimension of the problem and  $x_i^d$  defines as the position of the  $i$ th agent in the  $n$ th dimension.

In this algorithm, the agents of the solution are defined randomly by referring to the Newton’s Law of Gravity, a gravitational force where mass  $j$  acts on mass  $i$  at time,  $t$  as shown in following equation:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) X M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \tag{10}$$

where  $M_{pi}$  is the passive gravitational mass of agent  $i$ ,  $M_{aj}$  is the active gravitational mass of agent  $j$ ,  $G(t)$  is gravitational constant at time  $t$ , with  $\epsilon$  as a small constant and  $R_{ij}(t)$  is Euclidian distance between agent  $i$  and agent  $j$  as shown below:

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \tag{11}$$

The total force that acted on the agent  $i$  in a dimension  $d$  is defined as follows:

$$F_i^d(t) = \sum_{j \in kbest, j \neq i}^N rand_j F_{ij}^d(t) \tag{12}$$

where  $rand_j$  is a random number between interval [0, 1] and  $kbest$  is the set of first  $K$  agents with bigger mass and best fitness value.

To define the acceleration of the  $i$ th agent at the time  $t$  in the  $d$ th dimension, the Newton’s Law of Motion will be used. By referring to this law, the acceleration of the agent depends on its force and mass. So the acceleration of  $i$ th agent,  $a_i^d(t)$  can be defined as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{13}$$

where  $F_i^d(t)$  is the total force that acted on  $i$ th agent and  $M_{ii}(t)$  is the inertia mass of  $i$ th agent.

The next searching process involves finding the next velocity and next position of the agent. The next velocity of the agent referred to as a function of its current velocity added to its current acceleration where can be it described as follows:

$$v_i^d(t + 1) = rand_i X v_i^d(t) + a_i^d(t) \tag{14}$$

And the next position of the agent:

$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1) \tag{15}$$

where  $v_i^d(t)$  and  $x_i^d(t)$  is the velocity and position of  $i$ th agent in  $d$ th dimension at the time  $t$ .

The gravitational constant,  $G(t)$  is usually initialized at the beginning of the algorithm. This constant will be reduced according to time in order to control the search accuracy. This constant basically is a function of the initial value ( $G_0$ ) and time ( $t$ ) where:

$$G(t) = G(G_0, t) \tag{16}$$

and for this study, gravitational constant can be defined as:

$$G(t) = G(t_0) e^{(-\alpha X \frac{t}{t_{max}})} \tag{17}$$

where  $G(t_0)$  is the initial value,  $\alpha$  is an alpha constant,  $t$  is the current iteration and  $t_{max}$  is the maximum iteration.

The masses (gravitational and inertia) of the agents are calculated using the fitness function. By referring to the Newton’s Law of Gravity, agent with heavier masses becomes a more effective agent. In other words, a heavier mass has higher attractions and tends to move more slowly. For updating the gravitational and inertia masses stages, an assumption of equality for the gravitational and inertia mass will be applied as shown in following equation:

$$M_{ai} = M_{pi} = M_{ii} = M_i, i = 1, 2, \dots, N \tag{18}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{19}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \tag{20}$$

where:

$fit_i(t)$  = fitness value of the agent  $i$  at time  $t$

$best(t), worst(t)$  = strongest and weakest agent in the population respectively based on their fitness route.

For a minimization problem, the following are the equations:

$$best(t) = \min_{j \in \{1..m\}} fit_j(t) \tag{21}$$

$$worst(t) = \max_{j \in \{1..m\}} fit_j(t) \tag{22}$$

For a maximization problem:

$$best(t) = \max_{j \in \{1..m\}} fit_j(t) \tag{23}$$

$$worst(t) = \min_{j \in \{1..m\}} fit_j(t) \tag{24}$$

For the parameter selection of GSA-PID controller in this analysis, it was set to  $N = 20$ ,  $i = 100$ ,  $d = 3$  which represented  $K_P$ ,  $K_I$  and  $K_D$ ,  $G_0 = 100$ ,  $\alpha = 20$ , and  $\epsilon = 10$ . The same as PSO-PID controller, the range for PID controller parameters was selected from 0 to 100. For the optimization of problem solving by using GSA, every agent was placed at certain point of every search space which specific a solution to the problem and this step must be done at the beginning of the algorithm. Based on Equation. (14) and (15), the agents were rescheduled and their next positions were computed. After that, other parameters of the algorithm such as gravitational constant,  $G(t)$ , masses, ( $M$ ) and acceleration ( $a$ ) were defined via equations 16, 17, 18, 19 and 13 respectively and were updated on every cycle of time. The flowchart of the GSA is shown in Figure 2 [7], [13].

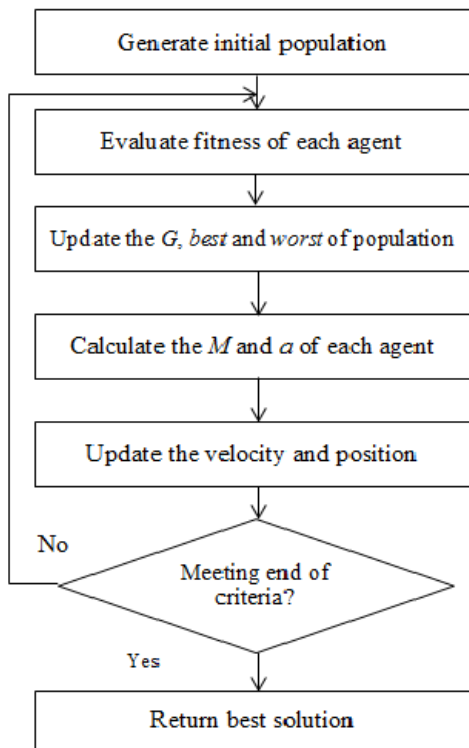


Figure 2 Flow Chart of GSA

#### 4.0 INTEGRATION GSA-PID CONTROLLER

There are many methods that provide PID controller tuning. A standard PID controller structure can also be defined as “three term” controller, where the transfer function is most often written in “parallel form” as shown in equation (25) and the ideal form as shown in equation (26) [14].

$$G(s) = K_P + \frac{K_I}{s} + K_D s \tag{25}$$

$$G(s) = K_P \left( 1 + \frac{1}{T_I s} + T_D s \right) \tag{26}$$

where:

- $K_P$  Proportional gain
- $K_I$  Integral gain
- $K_D$  Derivative gain
- $T_I$  Integral time constant
- $T_D$  Derivative time constant

The “three term” functionalities are described by the following:

- Proportional term – provide an overall control action proportional to the error signal through all pass gain factor.
- Integral term – reduce the steady state error through low frequency compensation by an integrator.
- Derivative term – improve transient response through high frequency compensation by a differentiator.

By representing the PID controller in terms of  $K_P, K_I$  and  $K_D$ , we can summarized that:

- Proportional controller ( $K_P$ ) will reduce the rising time, but will never eliminate steady state error.
- Integral controller ( $K_I$ ) will eliminate the steady state error but will make transient response worse, leadings to unstable system.
- Derivative controller ( $K_D$ ) will increase the stability of the system, reducing the overshoot and improving the transient response.

The basic structure of PID controller is visualized in Figure 3 below.

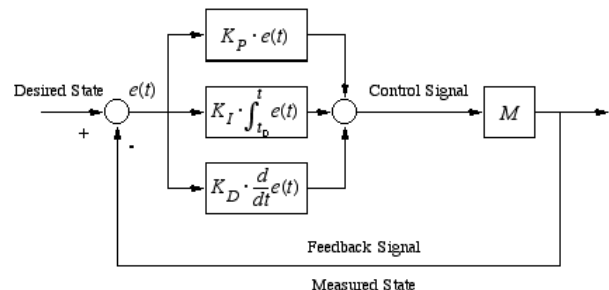


Figure 3 Basic structure of PID controller

In this section, integration between GSA and PID controller will be discussed. This GSA-PID controller will search for the optimal value for controller parameter  $K_P$ ,  $K_I$  and  $K_D$  using GSA algorithm. Each individual  $K$  contains three controller parameters  $K_P$ ,  $K_I$  and  $K_D$ . The searching procedure of proposed GSA-PID controller are shown as below.

- Step 1** Specify the lower and upper bounds of three controller parameters ( $K_P$ ,  $K_I$ ,  $K_D$ ) and initialize randomly the initial condition including searching for point,  $G_0$  and the number of iteration ( $iter_{max}$ )
- Step 2** Randomize number of controller parameters ( $K_P$ ,  $K_I$ , and  $K_D$ ). Check whether the random number fulfil the range or not.
- Step 3** Calculate the individual objective functions

- Step 4** Assign the weight of all objective and calculate fitness
- Step 5** Update  $G(t)$ ,  $best(t)$ , and  $worst(t)$  for the population.
- Step 6** Calculate the total masses ( $M$ ) and acceleration ( $a$ ) of the population
- Step 7** Find the new controller parameter position ( $X^{k+1}$ ) from velocity ( $V_i$ ).
- Step 8** Check stopping criteria

The flowchart of GSA-PID control system is shown in Figure 4 below.

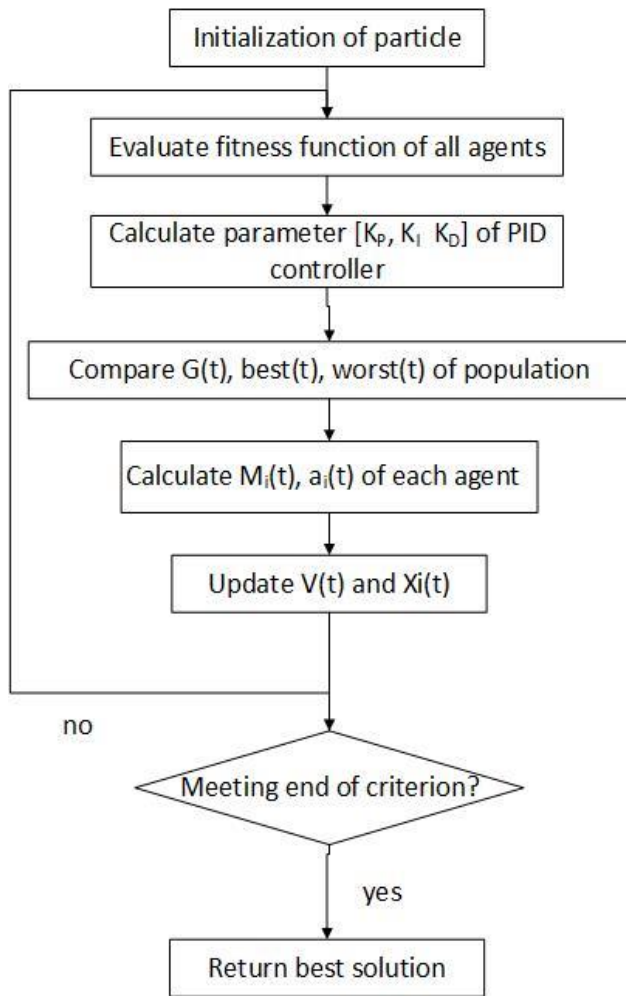


Figure 4 Flowchart of GSA-PID control system

**5.0 RESULT AND DISCUSSION**

To show the performance of the PID controller with a different optimization method (GSA and PSO) in controlling the concentration of substrate and dissolved oxygen of a waste-water treatment system, simulation studies were performed in this section. The simulations were done using MATLAB/ Simulink software with a specific setting of ODE45.

The structures of GSA and PSO techniques in PID controller tuning can be visualized in Figure 5 below and the plant modelling of activated sludge process is shown in Figure 6. The parameter tuning of a PID controller using GSA and PSO can be accomplished by choosing three parameters  $K_p, K_i$  and  $K_d$  such that

the output responses produced are much alike as the input response as shown in Figure 6 and Figure 7.

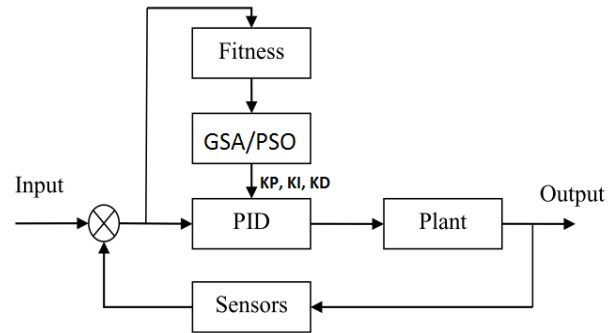


Figure 5 GSA/PSO implementation in PID tuning

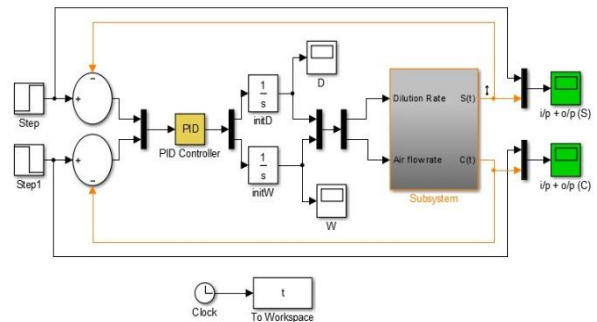


Figure 6 Plant modelling of activated sludge process

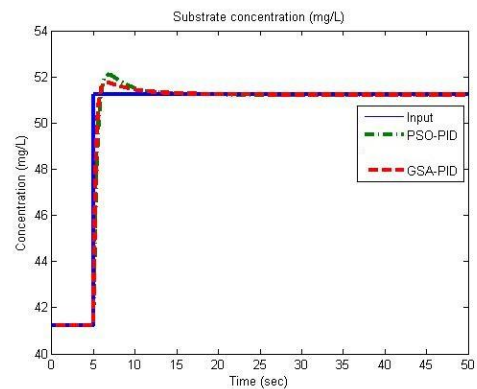


Figure 7 Substrate concentration comparison

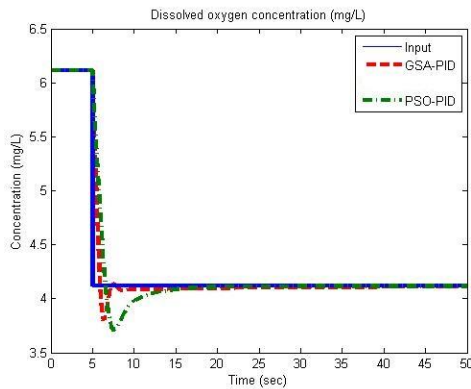
Figure 7 above represents the transient performance comparison between GSA-PID and PSO-PID for substrate concentration. Meanwhile, Table 1 below represents the transient response performance comparison for substrate concentration between GSA-PID and PSO-PID. The transient response was based on the closed loop performance of the plant, where the step input was injected into the plant. Based on the data in Table 1, GSA-PID controller produces the rise time of 0.51 seconds compared to 0.76 seconds for PSO-PID controller performance.

**Table 1** Transient response performance of GSA-PID and PSO-PID for substrate concentration

	Rise time, $t_r$ (s)	Settling time, $t_s$ (s)	Percentage Overshoot (%OS)	Steady State Error, (SSE)
GSA-PID	0.51	4.91	2.10	0.00
PSO-PID	0.76	5.78	7.29	0.00

In terms of settling time, GSA-PID produce faster settling time compared to PSO-PID controller where both produced 4.91 seconds and 5.78 seconds respectively. For the percentage of overshoot performance, GSA-PID produced 2.10% overshoot and it was better compared to 7.29% OS of PSO-PID. Both GSA-PID and PSO-PID produced 0% steady state error (SSE) results. Hence, from this analysis, closed loop performance for substrate concentration by GSA-PID controller was better compared to the performance of PSO-PID controller.

Figure 8 below represent the comparison performance between GSA-PID and PSO-PID in terms of dissolved oxygen concentration level. Meanwhile, Table 2 below represents the transient response performance for dissolve oxygen concentration. The transient response is for DO concentration which was also based on the closed loop performance of the plant. Based on the statistical data of both GSA-PID and PSO-PID responses, GSA-PID produced 0.70 seconds of rising time and it was faster compared to 1.35 seconds for PSO-PID.



**Figure 8** Dissolved oxygen concentration comparison

**Table 2** Transient response performance of GSA-PID and PSO-PID for dissolved oxygen concentration

	Rise time, $t_r$ (s)	Settling time, $t_s$ (s)	Percentage Overshoot (%OS)	Steady State Error, (SSE)
GSA-PID	0.70	6.93	3.87	0.00
PSO-PID	1.35	8.85	9.48	0.00

Meanwhile on the other hands, for the setting time analysis, GSA-PID produced a comprehensive 6.93 seconds compared to 8.85 seconds for PSO-PID which was slower. GSA-PID also produced lower percentage overshoot where GSA-PID produced 3.87% instead of 9.48% for PSO-PID performance. Both GSA-PID and PSO-PID produced 0% steady state error (SSE) result. From this statistical analysis, GSA-PID controller produced significant performance and was better compared to PSO-PID controller.

**Table 3** PID controller parameters value of GSA-PID and PSO-PID for dissolved oxygen concentration

	KP	KI	KD
GSA-PID	81.4920	4.9858	9.8337
PSO-PID	33.3847	8.7459	8.6895

Table 3 above represents the value of  $K_P$ ,  $K_I$  and  $K_D$  produced by GSA-PID and PSO-PID for both substrate and dissolved oxygen concentration analysis. Based on the results in Table 3 above, GSA-PID produced 88.4920 for  $K_P$ , 4.9858 for  $K_I$  and 9.8337 for  $K_D$ . Meanwhile, PSO-PID produced 33.3847 for  $K_P$ , 8.8759 for  $K_I$  and 8.6895 for  $K_D$ . Both substrate and dissolved oxygen concentration analysis used respectively the PID parameter of GSA-PID and PSO-PID because the system modelling were using centralized PID controller to control both responses. Based on the PID parameter value, it seems that GSA-PID produced larger value of  $K_P$  and followed by a much lower value for  $K_I$  and  $K_D$  compared to PID parameters produced by PSO-PID. From this point of view, it is believed that by producing larger value of  $K_P$  and a much lower value of both  $K_I$  and  $K_D$ , it may produce the best transient performance compared to other pattern of PID values.

### 6.0 CONCLUSION

From the results obtained in the previous section, the PID tuning based on Gravitational Search Algorithm (GSA) method has the capability to produce better performance in tuning PID controller parameters. Based on the transient analysis of substrate and dissolved oxygen concentration of waste-water treatment plant, GSA-PID provides better performance with faster rising time, faster settling time and lower percentage of overshoot compared to the performance of PSO-PID. A very fast time transient response of the closed loop of the system is needed due to the effectiveness of reducing losses such as maintenance cost and the efficiency of the system itself.

The characteristic of GSA where the parameter is based on gravitational performance produces accurate searching process than other optimization methods. Moreover, the control design of waste-water treatment process will improve, and hence, increase its potential to enhance environmental quality. Based on this point of view, GSA optimization is able to provide a good solution to the multi-variable optimization problems compared to PSO and other optimization algorithms.

### Acknowledgement

The authors wish to acknowledge the assistance of the Ministry of Higher Education, Malaysia and Universiti Teknologi Malaysia under the UTM-GUP Research Fund (Q.J130000.2623.09J03) and (R.J130000.7810.4F374)

### References

- [1] S. Gharghory and H. Kamal. 2013. Modified PSO for Optimal Tuning of Fuzzy PID. *International Journal of Computer Science*. 10(2): 462–471.
- [2] K. J. Astrom and T. Hagglund. 2006. *Advanced PID Control*. 1–460.
- [3] M. W. Iruthayarajan and S. Baskar. 2007. Optimization of PID Parameter Using Genetic Algorithm and Particle Swarm optimization. *IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES)*. no. Ictes. 81–86.
- [4] C. Ou and W. Lin. 2006. Comparison between PSO and GA for Parameters Optimization of PID Controller. *2006 International Conference of Mechatronics Automation*. 2471–2475.

- [5] L. F. Solihin, Mahmud Iwan, Tack and M. L. Kean. 2011. Tuning of PID Controller Using Particle Swarm Optimization ( PSO ). *Proceeding of the International Conference on Advanced Science, Engineering and Information Technology*.
- [6] B. Kumar and R. Dhiman. 2011. Optimization of PID Controller for liquid level tank system using Intelligent Techniques. *Can. J. Electr. Electron. Eng.* 2(11): 531–535.
- [7] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi. 2009. GSA: A Gravitational Search Algorithm. *Information Science (Ny)*. 179(13): 2232–2248.
- [8] Norhaliza Abdul Wahab. 2006. Multivariable PID control of an Activated Sludge Waste-water Treatment Process. 1(1).
- [9] R. Poli, J. Kennedy, and T. Blackwell. 2007. Particle Swarm Optimization. *Swarm Intelligence*. vol. 1, no. 1, pp. 33–57.
- [10] Y. Shi and R. Eberhart. 1998. A Modified Particle Swarm Optimizer. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*. pp. 69–73.
- [11] A. Asuntha and A. Srinivasan. 2014. Intelligent PID Controller Tuning Using PSO for Linear System. *International Journal of Innovative Scienens and Engineering Technology*. 1(5): 166–174.
- [12] M. F. Rahmat and M. Md Ghazaly. 2006. Performance Comparison Between PID And Fuzzy. *Jurnal Teknologi*. 45(D): 1–17.
- [13] E. Rashedi. 2012. Improving the Precision of CBIR Systems by Feature Selection Using Binary Gravitational Search Algorithm. *AISP 2012 - 16th CSI International Symposium on Artificial Intelligence and Signal Processing*. 39–42.
- [14] K. H. Ang, G. Chong, S. Member, and Y. Li. 2005. PID Control System Analysis , Design , and Technology. *IEEE Transaction of Control System Technology*. 13(4): 559–576.