VOL. 10, NO. 23, DECEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

EVALUATION ALGORITHM- BASED ON PID CONTROLLER DESIGN FOR THE UNSTABLE SYSTEMS

Erliza Binti Serri[1], Wan Ismail Ibrahim[1] and Mohd Riduwan Ghazali[2]
[1]Sustanable Energy & Power Electronics Research, FKEE UMP Pekan, Pahang, Malaysia
[2]Instrumentation & Control Engineering, FKEE UMP Pekan, Pahang, Malaysia
E-Mail: riduwan@ump.edu.my

ABSTRACT

PID controller consists of proportional, integral and derivatives controllers and it's widely used in industrial control system to provide optimal and excellent performance for any system. In term of controlling, unstable system exist when the system doesn't reach a steady-state value and will instead head towards infinity. This may cause damage to the system and might bid danger in certain systems. Evaluation algorithm develops to tune the PID controller for a better performance. In this project, the evaluating PSO algorithm based on PID controller design for unstable system was proposed. Particle Swarm Optimization (PSO) is computation method by simulation of swarms behaviour in performing their tasks. In the implementation of PSO in the PID controller, the swarm will travel to search the best value of parameter Kp, Ki and Kd. In this project, the stability impact of implementation of PSO in PID controller is being investigated. This implementation offers the stability effects to the unstable system by reducing the error and provides better performance of the system.

Keywords: PID controller, evaluation algorithm, PSO algorithm and unstable systems.

INTRODUCTION

Generally, this project is about implementing the evaluation algorithm in PID controller design. Proportional integral derivative (PID) controller is efficient and widely used as feedback control strategy. The PID controller has been successfully used in the process industries since the 1940s and remains the most often used algorithms today. The evaluation algorithm is implemented in the PID controller to improve the performance of PID controller. There are huge types of evaluation algorithm in the computational technique. In this project, PSO is choosing to use as the algorithm m method. This algorithm method will be implemented in the PID controller design that aims to stabilize the unstable systems.

Nowadays, people are accelerating for producing a quality product. In producing great and high quality product, the stable system is needed. However, some system may be unstable. In a control system, if the system is in unstable condition, its output will not reach a steady-state value and instead head toward infinity. This may cause damage to the system and might bid danger in certain system.

This project is aiming to stabilize the unstable system. To achieve that objective, this project combining both method to give a huge impact to the unstable systems. After confirming the control system is unstable, the PSO will generate the PID controller parameters and obtaining the three values that is Kp, Ki and Kd.

THE UNSTABLE SYSTEMS

The stability of a system is the most important part to be analyzed before designing a controller. The performance of a system can only be measured when it is in a stable condition. In term of controlling, if the system in unstable conditions, its output will not reach its steady state value and instead head towards infinity. It also may cause damage to the system, adjacent property and a human life.in this project, the stability of the system are determine by the root of transfer function in the S-plane. If the poles and zero of the transfer function is located at the right half S-plane. The system is unstable. Figure-1 below shows the S-plane for the unstable systems.

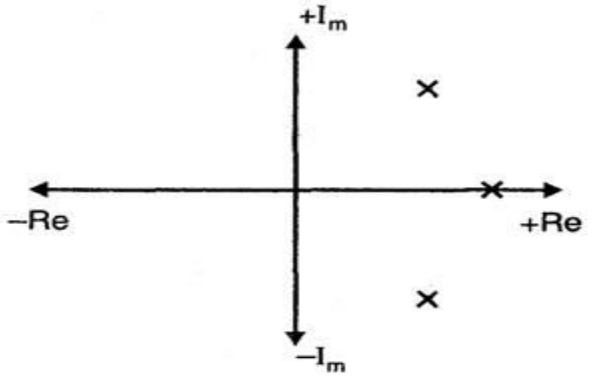

Figure-1. S-plane for the unstable systems.

PID CONTROLLER

The PSO algorithm perform their by searching the best parameter value of Kp, Ki and Kd. Generally, Kp will improve the systems by reducing the rise time. However, Kp never eliminates the steady-state error. Eliminating the steady-state error was done by Ki. But, Ki may make the transient response worse. Therefore, for Kd, it will effect, increasing the stability of the system, reducing the overshoot and improving transient response. The effect of increasing each of the controlled parameter Kd.

Traditionally, PID controllers were tuned manually using simple rules that dated back to Ziegler and Nichols in the 1940s. Besides, there many methods for tuning the PID controller. There are Ziegler-Nichols, Cohen-Coon, Bode's

Integral, Disturbance Rejection Magnitude Optimum, Pole Placement, Optimization Technique and the latest technique is the auto tuning method.

The rules were based on process experiments. The step response method is based on measurement of the open-loop step response. The frequency response method is based on a closed loop experiment where the system is brought to the stability boundary under proportional control. Unfortunately, the traditional rules gave systems with poor performance. Automatic tuning has increased the use of derivative action. It has even been said: "This controller must have automatic tuning because it uses derivative action."

Automatic tuning can be achieve in many ways. In rule-based methods that mimic an experienced instrument engineer, features of the closed-loop response are calculated and controller parameters are adjusted based on empirical rules [1]. For this project, we prefer to use the auto tuning method. Figure-2 below show the block diagram of PID controller.

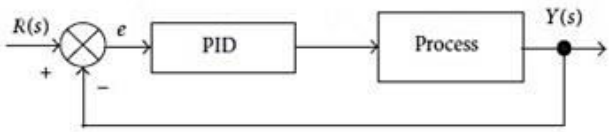

Figure-2. Block diagram of PID controller.

PARTICLE SWARM OPTIMIZATION (PSO)

Evaluation algorithm is a huge scope computation method that consists of many algorithms. In this project, the evaluation algorithm is PSO algorithm is being implement in the PID controller. PSO is a computational algorithm technique based on swarm intelligence. This method is motivated by the observation of social interaction and animal behaviors such as fish schooling and bird flocking. It mimics the way they find food by the cooperation and competition among the entire population [13]. A swarm consists of individuals, called particles, each of which represents a different possible set of the unknown parameters to be optimized. The swarm is initialized with a population of random solutions [2].

In a PSO system, particles fly around in a multi-dimensional search space adjusting its position according to its own experience and the experience of its neighboring particle. The goal is to efficiently search the solution space by swarming the particles towards the best fitting solution encountered in previous iterations with the intention of encountering better solutions through the course of the process and eventually converging on a single minimum or maximum solution [3]. The performance of each particle is measured according to a pre-defined fitness function, which is related to the problem being solved. The use of PSO has been reported in many of the recent works [4] in this field. PSO has been regarded as a promising optimization algorithm due to its simplicity, low computational cost and good performance [5].

IMPLEMENTATION OF PSO ALGORITHM IN THE PID CONTROLLER

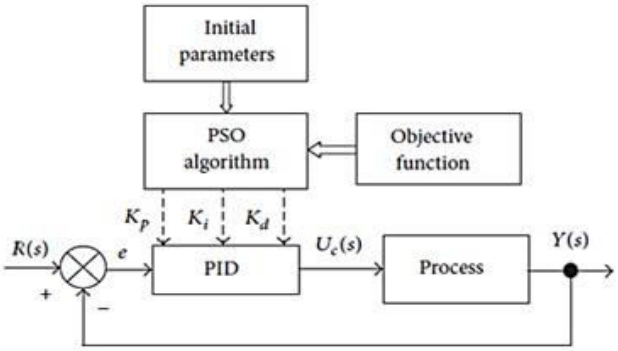

Figure-3. PSO-based PID controller design.

Figure-3 shows how the process of implementing the PSO algorithm in the PID controller. After obtaining the value of Kp, Ki and Kd, the value is inserted in the PID controller. The objective function interface runs the simulation. The simulation can be run either in the base workspace or the current workspace, that is, the workspace of the function calling sim, which in this case is the workspace of the interface.

PSO algorithm act as the searching agent, to search for the best value of Kp, Ki and Kd. Every particle has their own velocity and inertia in searching their best value. All the value obtains finally will be compared to each particle. The best value will be selected and finally the output is produced. When the simulation is completed, the outport block in the block diagram model puts the yout field of the object into the current workspace at the end of the simulation

RESULTS AND DISCUSSION

The unstable system

In this project, the transfer function that use is $T(s)=\frac{7}{s^3+2s^2+s+2}$. After determining the roots of the transfer function, the roots of the transfer function are s= -3.087, s= 0.0434 +j1.505 and s= 0.0434 -j1.505. All of the roots are located in the right-half of the S-plane. So, it is proved that the system is unstable. The S-plane for this transfer function is shown in Figure-4.

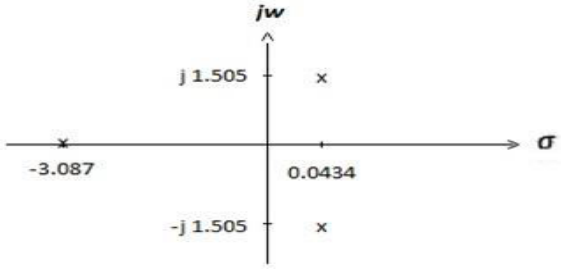

Figure-4. S-Plane for the transfer function of $T(s) = \frac{7}{s^3+2s^2+s+2}$.

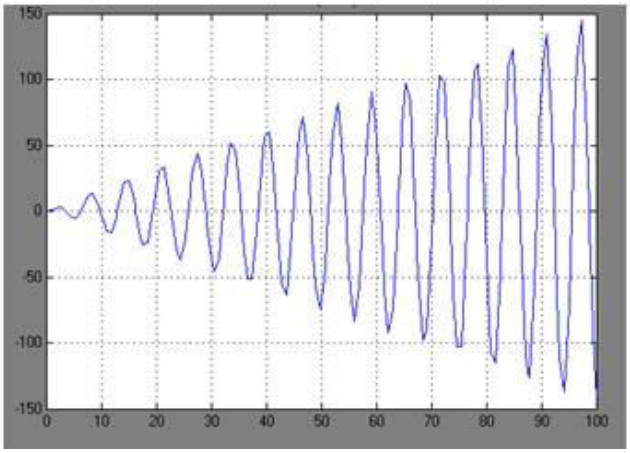

Figure-5. Output response for T(s) = $\frac{7}{s^3+2s^2+s+2}$.

Figure-5 shows output response for the transfer function T(s) = $\frac{7}{s^3+2s^2+s+2}$ that obtain from Simulink diagram shown in Figure-3. From output response, the output increased throughout the time until infinity. For the unstable system output shown in the figure, there is no analysis of settling time, rise time, overshoot neither peak time. All the value are infinity and can't be measured

Auto tuning PID controller to the unstable systems

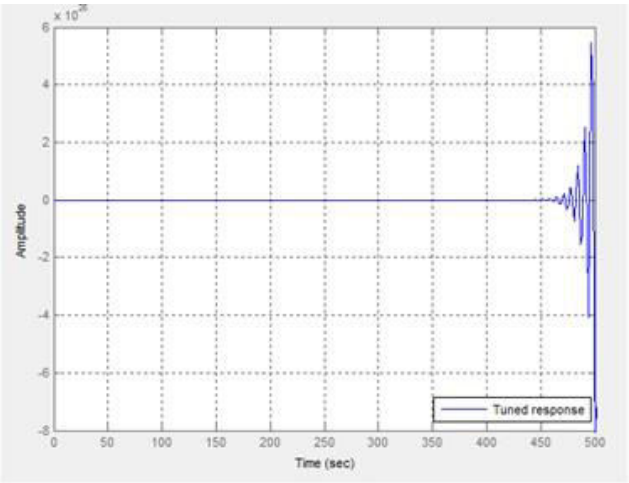

Figure-6. Step response for T(s) = $\frac{7}{s^3+2s^2+s+2}$ after auto tuning using PID controller.

Previous chapter already explains about description and the methodology about the auto tuning method. Figure-6 shows the step response result after auto tuning using the PID controller. Eventually, the auto tuning PID controller doesn't stabilize the unstable system. The step response is unable to analyse in term of peak time, settling time, overshoot and rise time because it's still in the unstable condition. This is because the performance of the system can only be measured when it in the stable condition.

PSO algorithm

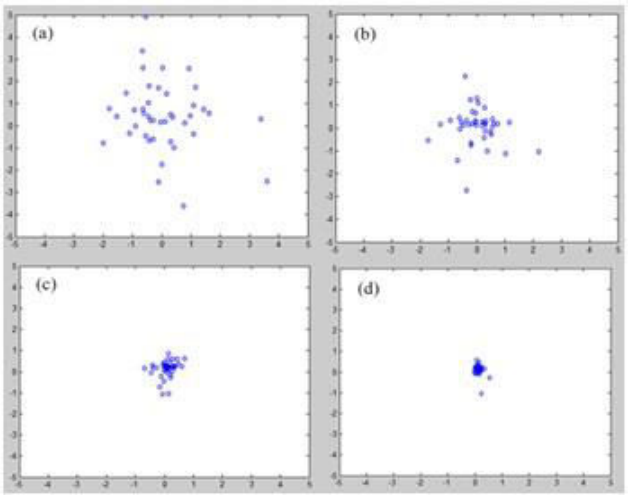

Figure-7. Generation of population in solution space.

Figure-7 (a) shows the generation of population inside solution space for tuning PID controller with PSO algorithm design for the unstable systems. The population consists of 50 particles using T(s) = $\frac{7}{s^3+2s^2+s+2}$. Figure 7 (b) shows the cluster formation as a result of an unbiased search by the particle for optimal value. Figure 7 (c) shows the unification of particle clusters enclosing best particle. While for the Figure 7 (d) show the appearance of the particle in the solution space after embedding the majority of the particle on the globally best one during the fine search phase.

PSO iteration

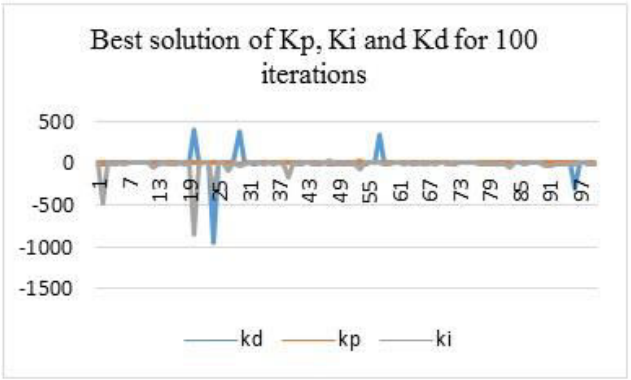

Figure-8. Generation of population in solution space.

In the previous chapter, the characteristic of the PSO was being discussed. The PSO initialized particle with random place and velocity. The figure below shows the analysis of random value of PSO algorithm in 100 iterations. In this paper, the simulation is considered to be attained of satisfactory fitness value which occurs with the maximum number of iterations as 100.

For each iteration the best among the 100 particles considered as a potential solution are chosen.

Therefore the best values for 100 iterations are sketched with respect to iterations, and are as shown in Figure-8.

After record all the 100th iterations, the value of Kp, Ki and Kd are different. It is synchronized with the PSO characteristic.

The implementation of PSO algorithm in the PID controller design

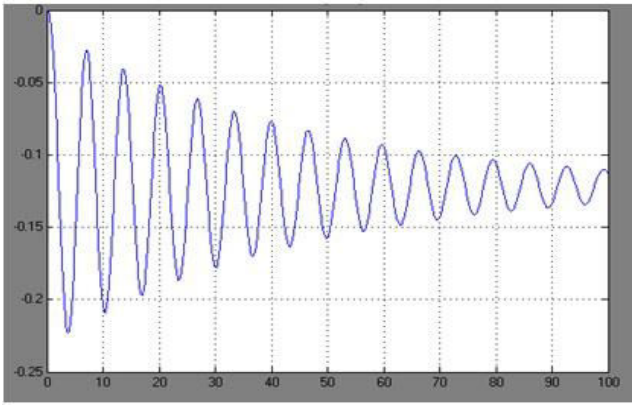

Figure-9. Output response for T(s) = $\frac{7}{s^3+2s^2+s+2}$ for P mode controller.

For the proportional mode only, it makes the control proportional to the error signal. Figure-9 above shows the output response graph of the system that using the P mode controller. The value of Kp that generate from the PSO algorithm -0.0310. The proportional mode provides fast response but doesn't reduce the offset to 0. It can't improve the output to achieve the steady state error. The output response shows that the proportional mode reduces the signal value to the negative value of infinite value at the unstable output. The amplitude increase due to the higher value of error. In conclusion, in the Proportional mode the output response not reaching the steady-state value and starting with the highest value and decreasing to the infinity.

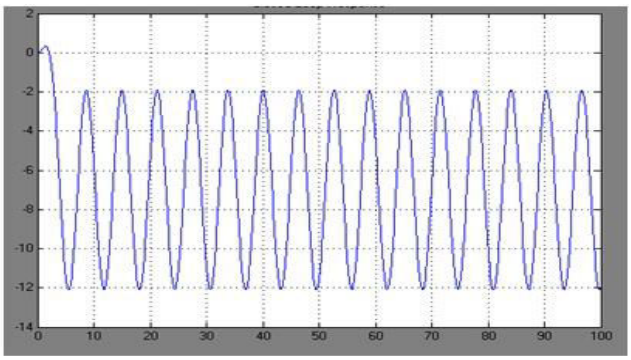

Figure-10. Output response for T(s) = $\frac{7}{s^3+2s^2+s+2}$ for PI mode controller.

For the proportional integral mode, it makes the control action proportionally for the error and make the control action until the magnitude of the error is reduced to zero to achieve the steady-state value. Figure-10 shows the output response graph of the system that using PI controller. The Kp and Ki value that obtain from generating the PSO algorithm is 0.5380 and -1.0269. At the PI mode, the proportional mode provides fast response, but doesn't the offset to zero, while for the integral mode reduces the offset to zero but provides relatively slow feedback compensation.

The combination of PI mode produces better signal than the P mode only in term of stability. For the output graph, the signal is initially decreasing to negative value and finally obtain at the steady-state value. For PI mode, the unstable system still doesn't reach the stability. The value of peak time, overshoot, steady-state, rising time and settling time.

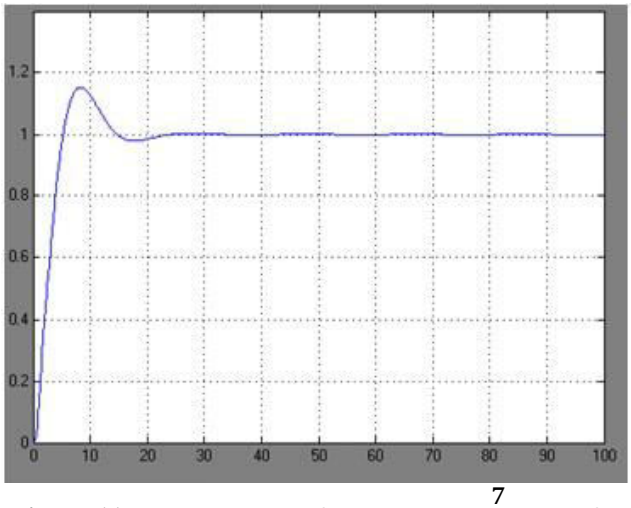

Figure-11. Output response for T(s) = $\frac{7}{s^3+2s^2+s+2}$ for PID controller.

Figure-11 is the output response graph for the Proportional Integral and Derivative controller graph. The value of Kp, Ki and Kd after generating from the PSO algorithm is 0.1738, 0.1559 and 0.9725. The three values are inserted in the PID controller and produce the above graph. The function of the proportional is used for accelerating the system, the integral is used to reduce the error while for the derivative mode is to reduce the oscillation of the system. Combining the three complete PID controller will stabilize the unstable system and become underdamped response.

From the graph, the rise time, peak time, settling time is 4s, 9s and 25s. The overshoot value is 1.18 amplitude. While, the steady state value is 0.

The other application of the unstable systems

The performance of the algorithm developed was tested with transfer functions of the unstable systems. The cost function here is the settling time in terms of system time constant which decides the performance of any industrial process.

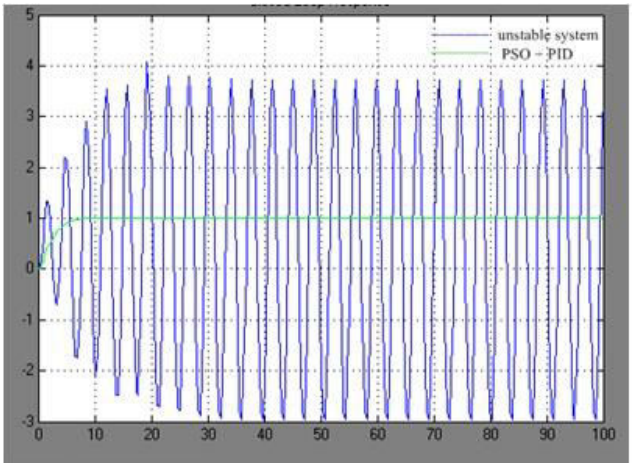

Figure-12. Output Response for T(s) = $\frac{11}{s^3+3s^s+4s+8}$.

The simulation result shows performance improvement of the unstable systems to stable systems. Using the PSO approach, global and local solution could be simultaneously found for better tuning of the controller parameter. The simulation output that contributes to PID parameter is Kp= 0.2367, Ki= 0.2885 and Kd= 0.7607. The performance of unstable system stability is shown in Figure-12.

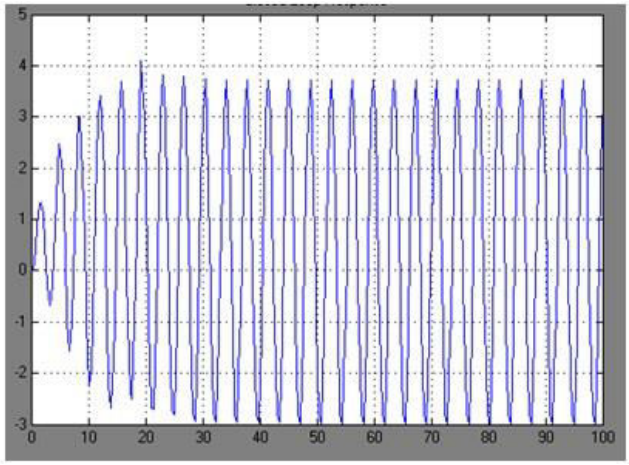

Figure-13. Output response for T(s) = $\frac{5}{s^3+3s^2+s+2}$.

Figure-13 shows the output response for the transfer functions of T(s) = 5/(s^3+3s^2+s+2). The graph obtain is following the unstable system rule that is the value doesn't reach the steady state value and keep increasing proportional with the time. If this system are obtain this kind of output, the system may damage due to unstable output obtain. It also may effects the operational cost. The system are starting its operation from zero and keep increasing.

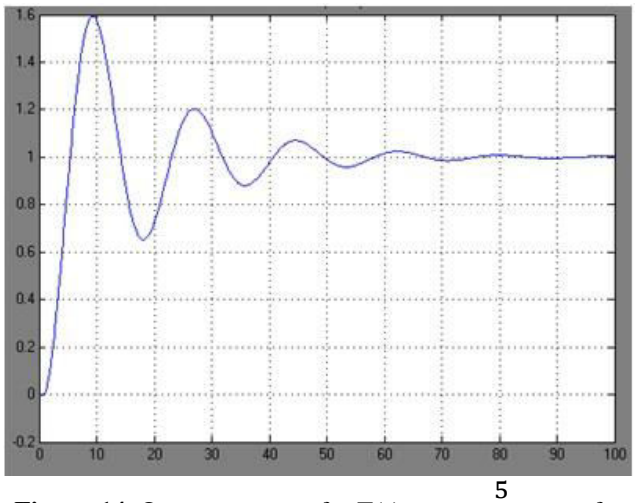

Figure-14. Output response for T(s) = $\frac{5}{s^3+3s^2+s+2}$ for PID controller.

Figure-14 shows the output response after tuning using the implementation of PSO in PID controller. The value of PID parameter after generating from PSO algorithm is -0.0738, 0.2787 and 2.0272. In the control system, the graph obtain is second order underdamped system. We can analyse the value of Tr, Ts, Tp, and overshoot by using Second-order underdamped response specifications at Figure-3.13. The value of Tr, Ts, Tp and overshoot value that obtain is 4s, 35s, 1s and 0.6 overshoot value. It is clearly prove that the implementation of PSO in PID controller stabilize the unstable system.

The closed loop PID controller cascaded with the process was tuned for the values Kp, Ki and Kd. Firstly, by using auto-tuning method and then by our modified PSO algorithm. Corresponding settling time was computed in both cases. In all the cases tested, the settling times obtained by PSO were much less than those values by the approximate method, as indicated in the Table-1.

Table-1. Comparison of Tr, Ts, Tp and Os obtained by auto tuning and PSO algorithm.

No.	Transfer function	Auto tuning PID				Implement PSO in PID			
		Tr (sec)	Ts (sec)	Tp (sec)	OS	Tr (sec)	Ts (sec)	Tp (sec)	OS
1.	$\frac{7}{s^3+2s^2+s+2}$	-	-	-	-	5	24	7	0.2
2.	$\frac{11}{s^3+3s^2+4s+8}$	38.7	69.3	1	0	7	14.2	1	0
3.	$\frac{5}{s^3+3s^2+s+2}$	135	239	1	0	4	35	1	0.6

Table-1 above shows that the comparison of peak time, settling time, peak time and overshoot obtained by the auto tuning PID and PSO algorithm. There are three the unstable system transfer function is being tested. For the first transfer function, T(s) = $\frac{7}{s^3+2s^2+s+2}$ the auto tuning PID doesn't produce any stability effect to this unstable system While, by using PSO algorithm, the

unstable system is stabilized with 24second of settling time. For the second and third unstable system transfer function, T(s) = $\frac{11}{s^3+3s^2+4s+8}$ and T(s) = $\frac{5}{s^3+3s^2+s+2}$. Both methods have stabilized the unstable systems . However, the auto tuning PID take longer time that PSO algorithm. It is clearly shown in the figure above that, the implementation of PSO in PID recorded the shortest time in term of rise time, settling time, peak time and overshot. The implementation of PSO in PID Controller only needs less than 10 second for rise time and less than 40 seconds settling time compared to auto-tuning method that needs longer time. The shortest time Tr, Ts, and Tp of the implementation of PSO in PID controller can improve the performance of some systems. The systems will operate smoothly without any delaying time to produce any product and to perform any task.

CONCLUSIONS

In this project, the Proportional Integral Derivatives controller is designed for the unstable system. The stability of the system are determine by the root of transfer function in the S-plane. The performance of the PID controller has improved by the existence of evaluation algorithm. The evaluation algorithm that chooses to implement in this project is PSO. The modified PSO algorithm implemented in the PID controller to show the impact of the unstable systems. Through th simulation testing and results, it is shown that PSO obtain the best value of Kp, Ki and Kd which can stabilize the unstable systems.

REFERENCES

[1] V. Rajinikanth and K. Latha. 2011. Bacterial foraging optimization algorithm based pid controller tuning for time delayedunstable systems. Mediterranean Journal of Measurement and Control. 7(1):197–203.

[2] Zhicheng X. 2010. A novel robust PID controller design method. Intnational Conference on Computer Application and System Modeling (Iccasm). pp. 332–337.

[3] Thomas B. B., Konstantinos E. P. and Michael N. Vrahatis. 2004. Analysis of particle swarm optimization using computational statistics. International conference on numerical analysis and applied mathematics ICNAAM. pp. 143-433.

[4] Jun Z., Tianpeng L. and Jixin Q. 2005. Application of particle swarm optimization algorithm on robust pid controller tuning. Springerlink-Verlag Berlin Heidelberg. pp. 948-957.

[5] Haibing H., Qingbo H., Zhengyu L. and Dehong X. 2005. Optimal PID controller design in PMSM servo system via particle swarm optimization. 31st Annual Conference of IEEE, Industrial Electronics Society. pp. 79-83.